Modelling Active Bio-Inspired Object Recognition in Autonomous Mobile Agents

Edgar Bermudez Contreras

Submitted for the degree of D. Phil.

University of Sussex

September, 2008

Declaration

I hereby declare that this thesis has not been submitted, either in the same or different form, to this or any other university for a degree.

Signature:

Contents

1 Introduction **24**

1.1 Structure overview . . . 26

2 Active object recognition and autonomous mobile robots **27**

2.1 Object recognition . . . 27

2.2 Object recognition models . . . 29

2.2.1 Computer vision approaches . . . 29

2.2.2 Biologically inspired approaches . . . 30

2.3 Active perception, embodiment, and situatedness . . . 34

2.3.1 Active vision and object recognition . . . 34

2.3.2 Embodied and Situated visual systems . . . 35

2.3.3 Movement and object recognition . . . 36

2.3.4 Temporal information and object recognition . . . 37

2.4 Controllers for autonomous visually guided mobile robots . . . 37

3 A first comparison: HMAX and RBF models in realistic conditions **39**

3.1 Introduction . . . 39

3.2 Visual system . . . 40

3.2.1 The Analysis module . . . 40

3.2.2 Classifier module . . . 47

3.2.3 The attentional and foveation mechanisms . . . 49

3.3 Model evaluation . . . 51

3.3.1 State of the art comparison . . . 51

3.3.2 HMAX implementation validation . . . 52

3.4 Comparison of the models in more realistic conditions . . . 56

3.4.1 Methods . . . 56

3.4.2 Results . . . 58

3.5 Conclusion . . . 62

4 Simulated Embodied Visual Systems **64**

4.1 Introduction . . . 64

4.2 Experiment 1: Using an active approach in a simple simulated agent. . . . 65

4.2.1 Methods . . . 65

4.2.2 Results . . . 67

4.3 Experiment 2: Increasing the complexity of the visual system. . . . 73

4.3.1 Methods . . . 73

4.3.2 Results . . . 77

4.3.3 Analysis . . . 79
4.4 Conclusion . . . 80

5 Active acquisition of visual information **82**
5.1 Introduction . . . 82
5.2 Methods . . . 82
5.2.1 Agent, arena and objects . . . 83
5.2.2 Training phase . . . 84
5.2.3 Testing phase . . . 87
5.3 Results . . . 87
5.3.1 Similarity maps . . . 89
5.3.2 Testing the models using movement trajectories . . . 94
5.4 Discussion . . . 99
5.4.1 Dimensionality . . . 101
5.4.2 The role of the BDM . . . 101
5.5 Conclusion . . . 102

6 Movement strategies during learning **104**
6.1 Introduction . . . 104
6.2 Methods . . . 105
6.3 Experiment 1: Movement strategies . . . 107
6.4 Experiment 2: Temporal information using the RBF model . . . 111
6.5 Experiment 3: Robustness of the RBF when using temporal information . . . 114
6.5.1 Changing the radius of strategy 3 . . . 114
6.5.2 Moving the centre of strategy 3 . . . 115
6.5.3 Using strategy 3 for training and the testing trajectory for testing. . . 116
6.5.4 Moving the centre of strategy 4 . . . 116
6.5.5 Considering interval timing for strategy 4 . . . 118
6.6 Experiment 4: Using more objects . . . 119
6.7 Discussion . . . 122
6.8 Conclusion . . . 123

7 A simple model of object recognition in the real world **125**
7.1 Introduction . . . 125
7.2 Methods . . . 126
7.2.1 Experimental set-up using the Gantry robot . . . 126
7.2.2 The visual system . . . 127
7.2.3 Movement strategies . . . 129
7.3 Results . . . 130
7.3.1 Experiment 1: Movement strategies and RBF model in the real world 130
7.3.2 Experiment 2: Using temporal information with the RBF model in the real world . . . 133
7.4 Discussion . . . 136

7.4.1 Differences between the real world and the simulated case 137
7.4.2 Exploitation of variation in the object views in the real world 137
7.5 Conclusion 138

8 Towards active selection of training views 139
8.1 Introduction 139
8.2 Methods 140
8.2.1 RBF versions 141
8.2.2 The classifier module 141
8.2.3 Movement strategies 142
8.3 Results 143
8.3.1 Reducing the complexity of RBF 144
8.3.2 Investigation of training views and model performance 146
8.3.3 Exploiting regularities in the environment through movement 151
8.4 Discussion 160
8.4.1 How could the reduced versions of the RBF model fully regain performance? 160
8.4.2 Towards active object recognition 160
8.5 Conclusions 161

9 General Discussion 163
9.1 Summary 164
9.1.1 Chapter 3 164
9.1.2 Chapter 4 164
9.1.3 Chapter 5 165
9.1.4 Chapter 6 165
9.1.5 Chapter 7 166
9.1.6 Chapter 8 166
9.2 Future work 167
9.3 Final conclusions 168

List of Figures

1.1 Brighton seafront on a Sunday morning. 24

2.1 Ventral and dorsal pathways in the visual cortex. The activity of the ventral pathway is generally associated with the identification of objects while the dorsal pathway is commonly associated with the localisation and actions related to objects in space (image adapted from wikipedia). 31

2.2 Hierarchical structure in the HMAX model. (adapted from (Riesenhuber and Poggio, 1999b)) 33

3.1 The visual system consists of the Analysis module (a) which can be either the HMAX or RBF model, and the Classifier module (c). The object recognition process starts with visual information coming into the visual system, then the Analysis module processes this information and outputs a vector (b) that the Classifier module analyses and characterises into the set of known objects. The output of the classifier module is the identifier (label) with the highest response for the input image. 40

3.2 RBF model: The representation of an object view is given by the filtered incoming visual information using low-pass filters of different sizes and orientations. Three sampled regions are shown here. 41

3.3 Example of filters with a particular size and four different orientations. 42

3.4 HMAX model: The representation of an object view is given by a more complex processing using a hierarchical structure. Figure adapted from (Riesenhuber and Poggio, 1999b) 43

3.5 Illustration of HMAX layers shown in figure 3.4: (a) original picture. (b) image filtered after S1 layer, using a filter sensitive to horizontal segments. (c) image after layer C1, using the max pooling operation. (d) image after layer S2, applying the gaussian smoothing and subsampling. Finally, by taking the max operation over a combination of scales and sizes, a vector is obtained as a result of the max pooling operation. 46

3.6 View Tuned Unit (VTU): each training view c_i is the centre of a Gaussian function. The more similar a vector x is to a centre, the stronger the response of the unit. The output of the VTU, $y = \sum_i W_i G(c_i, x)$ 47

3.7 Classifier module: each object is represented by a View-Tuned unit (VTU). When a vector is analysed, the output of the classifier module is the maximum of all VTU responses. 49

3.8 Blob detection mechanism. A) Image B) edges detected C) dilated edges D) Filled holes E) Connected components (detected blobs) F) Resized selected blob (in this case the largest area criteria was applied to select the blob). 50

3.9 HMAX implementation output comparison. The lines represent the mean of the HMAX implementation outputs over 100 random 32×32 pixel black and white images from the COIL-100 library. The green line represents the output of HMAXm (one filter size per band and first derivative of Gaussian filters), the blue line represents the output of the original version of HMAX (HMAXo) and the red line represents the HMAX$_m$ implementation but using the same type of filter as in the original version. 53

3.10 Difference in the output of HMAX versions. The first column is $||\text{HMAX}_o(i)$ - $\text{HMAX}_m(i)||$, the second is $||\text{HMAX}_o(i)$ - $\text{HMAX}_m(j)||$, the third one is $||\text{HMAX}_m(i)$ - $\text{HMAX}_m(j)||$ and the fourth one is $||\text{HMAX}_o(i)$ - $\text{HMAX}_o(j)||$, where $i \in I$ and $j \in J$ and $|| \cdot ||$ is the Euclidean norm. 54

3.11 Coil library examples. Ten views of different objects of the COIL-library. 54

3.12 Image Set: object 1 (rubber wheel), object 2 (usb adaptor), object 3 (phone connector), object 4 (pencil sharpener), object 5 (deck of cards), object 6 (light sensor), object 7 (IR sensor). 8 views for each object. 57

3.13 Translation invariance experiments. Scenario I: Uniform background, no foveation, 7 objs, 1 view. Scenario II: Non-uniform background, foveation, 7 objs, 1 view. Scenario III: Non-uniform background, foveation, 7 objs, 8 views. Scenario IV: Non-uniform background, foveation, 7 objs, 8 training views, testing in various positions, foveation. 59

3.14 Scale invariance experiments: scenario I, scenario II, scenario III, scenario IV (previously described in figure 3.13). 61

4.1 Agent body: two wheels on each side driven by independent motors. Two sensors placed at $\pm\pi/4$ radians from the line of orientation of the body. For directional sensors the agent can only perceive light coming from objects in the grey area. For panoramic sensors the light can be perceived from any direction. 66

4.2 Neural controller: a CTRNN with 8 nodes. Neurons 0 and 4 are the sensor nodes, neurons 2, 3, 6 and 7 are fully connected interneurons and neurons 1 and 5 are the motor neurons. The width of each arrow represents the strength of the connection (weight). The solid lines in the arrows represent excitatory connections and the dotted lines in the arrows represent inhibitory connections. 67

4.3 Dynamics of an evolved controller of 8 neurons using panoramic sensors (left panel). Neurons 0 and 4 are sensor neurons, neurons 1 and 5 are motor neurons and neurons 2, 3, 6 and 7 are interneurons. (A) Positions of an evolved agent during a test run. The object (target) is placed in the center of the arena (0,0). (B) Distance between the agent and the object during the test run (timestep vs distance). 68

4.4 Long term steady state of the neural controller: the agent was fixed in a position facing right (indicated by a line) and the object was moved around it. After 50 timesteps the activation of each neuron is stored. Red regions represent 1 in the output of the neuron when the object is in that position and blue regions represent 0 in the output of the neuron when the object is in that position. 69

4.5 Neural activity of an evolved controller of 8 neurons using directional sensors during a test trial. Neurons 0 and 4 are sensor neurons, neurons 1 and 5 are motor neurons and neurons 2, 3, 6 and 7 are interneurons. (A) Positions of an evolved agent during a test run. The object (target) is placed in the centre of the arena (0,0). (B) Distance between the agent and the object during the test run. 70

4.6 Neural activity of an evolved controller using 6 neurons and directional sensors during a test. Neurons 0 and 3 are sensor neurons, neurons 1 and 4 are interneuron and neurons 2 and 5 are motor neurons . (A) Positions of an evolved agent during the same test. The object (target) is placed in the center of the arena (0,0). (B) Distance between the agent and the object during the test run. 70

4.7 Simplified neural network: 6 nodes. Only two interneurons. The dotted line arrows represent inhibitory connections and solid arrows represent excitatory connections. The width of the arrows is proportional to the strength of the connection. 71

4.8 Average fitness of the population for the first 500 generations for different controllers and types of sensors. Controllers with 8 neurons and panoramic sensors (8NP), controllers with 8 neurons and directional sensors (8ND) and controllers with 6 neurons and directional sensors (6ND). 72

4.9 Ambiguous situation: the activation in the sensors when the object O is in front of the agent, is equivalent to the activation generated from the object O'. The distance from O to the sensors is the same as the distance from O'. That is A = A' and B = B'. 72

4.10 Simulated visual system. The visual field of the agent is a region of 512×32 pixels. L is the distance from the object to the left edge of the visual field and R is the distance from the object to the right edge. The inset A in the figure shows the detected blobs (from a distance of 2.5 to the dark object and 3.0 units to the light object) containing the light and dark objects respectively. In this example, the dark object is the largest and so the sensor neurons will respond to this object. . . 74

4.11 Visual field of the SSA: the object (O) can only be sensed if it is within the dark brown region. This region is limited by two lines extending from the center of the agent at ±45 deg from the agent's orientation and a width of V. L and R are the distances between the object and the left and right edges of the visual field, respectively. 75

4.12 Controller. Neurons: 0 and 4 are location sensors; 8 and 9 are colour sensors. Neurons 1,2, 5 and 6 are fully connected. Neuron 3 is the left motor neuron and neuron 7 is the right motor neuron. Note that the colour sensor neurons 8 and 9 were *not* used for the object approaching task. 76

4.13 Object approaching by an SSA. [A] shows the neural activity during a test trial of 800 time-steps. [B] shows the distance between the agent and the object during the trial and [C] shows the distance between the agent and the object during the trial. 77
4.14 Object approaching performed by an RSA using the evolved controller shown in figure 4.13. [A] shows the neural activity during the test trial of 800 time-steps. [B] shows the trajectory of the agent during the trial and [C] shows the distance between the agent and the object during the trial. 78
4.15 Object discrimination performed by an SSA. [A] shows the neural activity during a test trial of 800 time-steps. [B] shows the trajectory of the agent during the trial and [C] shows the distance between the agent and the object during the test trial. 78
4.16 Object discrimination performed by an RSA using the evolved controller shown in figure 4.15. [A] shows the neural activity during a test trial of 800 timesteps. [B] shows the trajectory of the agent during the trial and [C] shows the distance between the agent and the light object during the trial. 79

5.1 View Tuned Unit (VTU): each view vector c_i is the centre of a Gaussian function. The more similar a vector x is to a centre, the stronger the response of the unit. . 84
5.2 (A) Visual field of the agent: shows object 1 in the field of view. (B) Sample views of object 1 and object 2: object 1 is a rounded object so it does not have a significant variability to rotation, in contrast, object 2 has a significantly higher variability to rotation due to its vertical inclination. 84
5.3 Training trajectory. The agent follows a circular trajectory while collecting the training views. The number of views used (8 or 16) for each object determines the positions around the object. Solid lines show 8 different positions where the snapshots (training views) are taken. Similarly, dotted lines show the case where 16 training views are taken. 85
5.4 Training views for scenario 1. 8 views per object. Object 1 is a teapot and object 2 is a bolt-like object. Every view is 80 $\times$60 pixels. 86
5.5 Training views for scenario 3. 16 views per object. The sizes of the images are the same as the ones presented in the previous figure. 86
5.6 Trajectories followed by the agent during the testing phase. In trajectory 1, the agent approaches the two objects following an arc trajectory (dotted line). The objects are within the visual field in the shadowed regions in the trajectory. The initial position of the agent is (3,0) and the positions of the objects 1 and 2 are (0,4) and (0,-4) respectively. In trajectory 2, the agent approaches the objects following a straight line (dotted line). The object is always within the field of view. The initial position of the agent is (0, 0.5) and the position of the object is (0, 4). . . . 87
5.7 Average performance (%) of the RBF and HMAX models over the four scenarios when tested using trajectories 1 and 2. On average, the performance of the RBF is better than the performance of the HMAX model. 89

5.8 Similarity map diagram. Every point in the similarity map (i, j) represents the distance (Euclidean norm) between the views i and j (in this case, after being processed by the RBF model and using 8 views per object). This diagram also shows the regions in a similarity map. 90

5.9 Similarity map for scenario 1: (A) shows the similarity map for the images. (B) shows the similarity map of the RBF views (C) shows the similarity map for HMAX views. The blue tones represent high similarity while the red tones represent high disimilarity (difference). 91

5.10 Similarity map for scenario 2: (A) shows the similarity map for the images. (B) shows the similarity map of the RBF views. (C) shows the similarity map for HMAX views. For this scenario, the noise in the centroid of the blob detected makes the maps less uniform (compared to scenario 1). 92

5.11 Similarity map for scenario 3: (A) shows the similarity map for the images. (B) shows the similarity map of the RBF views (C) shows the similarity map for HMAX views. 92

5.12 Similarity map for scenario 4: (A) shows the similarity map for the images. (B) shows the similarity map of the RBF views (C) shows the similarity map for HMAX views. 93

5.13 ((A) Recognition signals of the two models for trajectory 1 for scenario 1. (B) Difference between the recognition signal of object 1 and object 2 (object 1 - object 2) for scenario 1. During the first period, both models classify the views of object 1 correctly except for the HMAX model at the end of the first period (where the blue line is negative). For the second period, the HMAX model misclassifies the views of object 2 most of the time. In contrast, the RBF model performs correctly most of the time. 95

5.14 (A) Recognition signals of the two models for trajectory 1 for scenario 2. The signals of both objects become more similar, compared to the signals in scenario 1. (B) Difference between the recognition signal of object 1 and object 2 (object 1 - object 2) for scenario 2. During the first period, the RBF model classifies the test views of object 1 correctly (red line is positive). In contrast, the HMAX model misclassfies them most of the time (blue line negative). For the second period, the RBF misclassifies the test views of object 2 (positive red line), while the HMAX model classifies them correctly (negative blue line). 96

5.15 (A) Recognition signals of the two models for trajectory 1 for scenario 3. With the absence of noise in the blob detection mechanism and more training views, the difference between the recognition signals is larger than in the previous cases. (B) Difference between the recognition signal of object 1 and object 2 (object 1 - object 2) for scenario 4. During the first period, the recognition signal for object 1 is higher than the signal for object 2 for both models. However, for the second period, only the RBF signal for object 2 is higher than the signal for object 2. . . 96

5.16 (A) Recognition signals of the two models for trajectory 1 for scenario 4. (B) Difference between the recognition signal of object 1 and object 2 (object 1 - object 2) for scenario 4. During the first period of the test, both models classify the test views correctly, except at the end of the first period for the HMAX model (negative blue line). For the second period, the HMAX model becomes chaotic. The RBF model classifies the test views most of the time. 97

5.17 Similarity maps for the test phase when the agent is following trajectory 1. The left column represents the maps for the RBF model and the right column, the maps for the HMAX model. The different rows correspond to the scenarios in which the models were trained. 98

5.18 Model activity during the testing phase when the agent was following trajectory 2 for the four scenarios. Given that the output of the models is deterministic, the model activity shown in this figure is the same every time the agent approaches the object in the same way. 100

6.1 (A) Visual field of the agent: shows object 1 in the field of view. (B) Sample views of object 1 and object 2: object 1 is a rounded object so it does not have a significant variability with respect to rotation, in contrast, object 2 has a significantly higher variability with respect to rotation due to its vertical inclination. 105

6.2 Movement strategies. While following the movement strategies, the agent takes snapshots at uniform intervals. Strategy 1: the agent approaches the object in a straight line. Strategy 2: the agent passes the object following a straight line. Strategy 3: the agent circles the object with a fixed radius. Strategy 4: the agent spirals the object. The testing trajectory consisted of two phases which correspond to the grey segments. In the first period object 1 was within the field of view and, in period 2 object 2 was within the field of view. 106

6.3 Movement strategies and model performance. The performance of the RBF model increases when the movement strategies allow it to exploit the rotational information during training. In contrast, the HMAX model performance decreases when the model is exposed to multiple rotational views during training in strategies 3 and 4. The performance of the models refers to the number of times the model has a correct guess over the test phase (only averaged over the total number of presentations during the test phase). Since this number depends on the presentation of object views which are deterministic (the same views will be presented every time the agent follows the corresponding trajectory) and the input-output mapping of the models is deterministic, the bars in this figure do not consider any statistical measure of variance. Note that chance level is 50%. 108

6.4 RBF and HMAX models activity during the test phase using strategy 3. When the movement strategy provides multiple points of view during the learning phase, the RBF can have a close match between the training and the test views. In contrast, the HMAX model decreases its discriminability when more points of view are considered. Period 1 represents the time when object 1 is within the visual field. Period 2 is the time when object 2 is within the visual field. 110

6.5 Similarity maps of the models using strategy 3. The darker the regions in each map, the more similar the corresponding views. For the RBF map there is an obvious darker region in the left lower area (corresponding to the views of object 1) for the first period, and a smaller darker region in the right upper area (corresponding to the views of the object 2). In contrast, for the HMAX similarity map dark areas appear during both periods for views associated with both objects. 111

6.6 RBF recognition signal (model activity) using the DBCV and trained using strategy 3 and tested in the same trajectory with randomised and normal ordered training views. The left column figure shows the RBF recognition signal for normal conditions and the right column figure shows the recognition signal for random ordered training views. 113

6.7 RBF model activity when trained using strategy 4 and tested using the testing trajectory. Left column: normal order of the training views. Right column: random order of the training views. 113

6.8 RBF model activity for different radii of strategy 3. The left column shows the activity when using ordered training views. The right column shows the activity when the order of the training views was randomised. The first row shows the activity when using the same radius during training and testing. The middle row shows the activity when the radius of the testing strategy was increased to 4. Finally the bottom row shows the activity when the radius is 6. 114

6.9 RBF model activity when the centre of the strategy 3 was moved during the test phase. The left column shows the activity when using ordered training views. The right column shows the activity of the model when the order of the training views was randomised. The figures in the first row show the activity when placing the centre at (0, 4.2). The figures in the middle row show the activity when the centre was placed at (0, 4.8) and the figures in the bottom row show the activity when the centre was placed at (0, 6). 115

6.10 RBF model activity for the testing trajectory when using strategy 3 during training. The left column shows the activity when using ordered training views. The right column shows the activity when the order of the training views of object 2 are randomised. 116

6.11 RBF model activity when trained and tested using strategy 4 and when the centre of the trajectory is moved during the test. The left column shows the activity when ordered views were used during training. The right column shows the activity when views were randomised during training. The top row shows activity when the center of the testing trajectory was at (0, 4) (as in learning). Middle row shows activity when the centre of the trajectory was at (0, 4.8) and the bottom row shows activity when the centre was at (0, 6.0). 117

6.12 RBF model activity during testing trajectory. The model was trained using strategy 4: In the top row, the interval between each view is 10 time steps during training and testing. In the bottom row, the interval was 3 time steps during training and 1 time step during testing (continuous). The left column shows the activity when the model is trained using ordered views and the right column shows the model activiy when using randomised training views. 118
6.13 Examples of views of the objects. Object 1: teapot, object 2: bolt1, object 3: bolt2, object 4: textured house, object 5: extinguisher, object 6: webcam. 119
6.14 Model response during the testing phase. The first row shows the activity of the RBF model when a single view presentation (SVP) was used. The second row shows the model activity when DBCV was used. The left column shows the model activity when the model was trained using strategy 3 and tested using the testing trajectory. The right column shows the model activity when the model was trained using strategy 3 and tested using strategy 4. 120
6.15 Comparison of the performance of the RBF model for DBCV and SVP conditions. The percentage corresponds to the average number of the total correct classifications during the test phase. For the trajectory 1 (left graph) the total number of presentations is 110 (55 in period 1 and 55 in period 2). For strategy S4, the total number of presentations is 200. Since the model is deterministic, the bars only represent the average of the correct guesses over the total number of views (presentations) during the test trial. 120
6.16 RBF model activity during the testing phase and using strategy 4 during training. Object 1 was present in period 1 and object 2 was present in period 2. This time the model was trained using 6 objects. 121
6.17 RBF model activity comparison between the cases when using 3 or 6 objects during training when using SVP and DBCV. For this experiment, object 3 was present in the arena during the test phase. 122

7.1 Gantry robot and the seven objects used for the object recognition tasks. At the end of the mechanic arm, there is a panoramic camera. Inset: 1700 × 2900 mm arena. The black circle represents the circular buffer where the object was placed. 127
7.2 Example of a panoramic image and the corresponding unwrapped and cropped image. 128
7.3 Recognition maps for every strategy using the RBF with SVP. The left column shows the four movement strategies to collect the training views. 131
7.4 Advantageous zone. Recognition map when using SVP and strategy 1. 132
7.5 Recognition map using temporal information. The circles and solid lines figure shows the order in which the training views were considered when calculating the DBCV. The right column in the figure shows the recognition maps when the model was trained using each strategy and tested in every valid position of the arena. . . 135
7.6 Advantageous zone for the recognition maps using DBCV. Recognition map when using DBCV and strategy 4 during training. 136

8.1 Movement strategies used to collect the training views. A) Movement strategy T1: the agent approaches the object in a straight line. B) Movement strategy T2: the agent passes in front of the object in a straight line. C) Movement strategy T3: the agent circles the object. D) Movement strategy T4: the agent spirals around the object. 142
8.2 Spiral strategy. 100 positions were generated spiraling around the object (circle in the centre). 143
8.3 Performance (%) of the different RBF reduction implementations. The performance of RBF_A, RBF_B and RBF_C are represented by columns A, B and C, respectively. The performance corresponds to the average number of correct guesses for every position in the arena using the four training movement strategies (averaged across all 4 movement strategies). The error bars show the standard deviation over the movement strategies. 144
8.4 Similarity map of the training views using RBF_A and RBF_C. The similarity between views v_i and v_j is defined as $\| v_i - v_j \|$ where $\| \cdot \|$ is the Euclidean norm. The red colour in the map indicates the lowest similarity and blue colours indicate the highest similarity between the training views. The red areas in the RBF_A similarity maps are yellowish and blueish in the RBF_C, showing the reduction in the specificity in the reduced versions of the RBF model. The similarity map using RBF_B (not shown in this figure) shows an intermediate state between RBF_A and RBF_C. 145
8.5 Difference of similarity map. In this map, every point represents the difference between the map RBF_A and the map RBF_C after normalising their values by the maximum view difference for each object and movement strategy (from figure 8.4). Blue represents low differences (min value = -0.15), white represents neutral difference (zero) and red represents larger differences (max value = 1.6). 146
8.6 Total number of correct classifications by the RBF_A when the training views were collected using movement strategies T1, T2, T3 and T4 across all objects. 147
8.7 Total number of correct guesses for each object for the RBF_A model when using movement strategies T1, T2, T3 and T4. Objects 2 and 6 are the ones with the lowest performance. Note that the quantities represented by each colour are independent and they are not meant to represent a cumulative plot. 147
8.8 Examples of training views of object 4 (squirrel) using movement strategies T1, T2, T3 and T4. 148
8.9 Total number of correct guesses for each object and each movement strategy (T1, T2, T3 and T4) for the RBF_A model. Note that the quantities represented by each colour are independent and they are not meant to represent a cumulative plot. . . 149
8.10 Total number of correct classifications by the RBF_A model using movement strategies T1, T2, T3 and T4, when the blobs were manually corrected. 149

8.11 Map of correct guesses for each movement strategy. The darkest blue colour represents 0 correct classifications and the darkest red colour represents 7 correct classifications at a given location in the arena (one per object). The white points represent the locations in the arena where the training views were collected for each movement strategy. 151

8.12 Order of the views along the spiral movement strategy. The circles represent positions along the spiral trajectory (dotted line) where views were collected. The normal order is anticlock wise, the first view (1st) is the one closer to the object and the last one (100th) is the one further away from the object. The inverse order is in the opposite direction, clock wise, starting with the further position in the trajectory from the object. 152

8.13 Examples of the training views selection methods. The dotted lines represent a region in the arena and the intersections of these lines represent a valid position in the arena. The solid line represents a segment of the trajectory of the spiral movement strategy. The interval based method (left) selects the training views at fixed intervals. The solid circles represent positions selected to collect training views and non-solid circle represents a non-selected position (interval 1 in this case). The threshold based method (middle) measures the similarity between the current view (black solid circle) and the next view (grey solid circle). The non-solid circle represents a previous point in the spiral where the similarity was not lower than the threshold. Finally, the neighbourhood based method (right) calculates the similarity values between the current view in the spiral movement strategy and its neighbours (in the arena) in the four cardinal directions. 152

8.14 Total number of correct classifications by the RBF models when the training views were selected using a fixed interval strategy. Six intervals were used in both, normal and inverse order for each RBF version. The same classifier parameters ($\sigma = 1.8$) were used for RBF_B and RBF_C not only for this training view selection method but also for the threshold and neighbourhood based methods as well. 153

8.15 Similarity map of the views in the extended spiral movement strategy for the 7 objects (columns) and the RBF_A, RBF_B and RBF_C models (rows A, B and C respectively). Red colour represents low similarity between the views and blue colour represents high similarity between the views. 153

8.16 Positions where the training views were collected in the arena when using the threshold based method and the RBF_A, RBF_B and RBF_C models for all objects. The intensity of the colour of blue circles (RBF_A), red squares (RBF_B) and green triangles (RBF_C) represents the number of views selected in that position in the arena. The more intense the colour, the more views were selected in that position. The black dots represent the positions of the views using an interval of 3 views in inverse order. 156

8.17 Total number of correct classifications for RBF_A, RBF_B and RBF_C models when different methods for selecting training views were employed. 158

8.18 Example of locations where the training views were collected for object 1 using the RBF_A model for the different view selection methods. Red dots represent the locations where the views were collected and the blue dots represent the object location in the arena. (A) Interval based method with interval of 1 view in normal order. (B) Interval based method with interval of 3 and inverse order. (C) Threshold based method. (D) Neighbourhood based method. 158

8.19 Positions where training views were collected using the neighbourhood based method for the three versions of the RBF model and every object. The intensity of the colour of blue circles (RBF_A), red squares (RBF_B) and green triangles (RBF_C) represents the number of views selected in that position in the arena. The more intense the colour, the more views were selected in that position. The black dots represent the positions of the views using an interval of 3 views in inverse order. . 159

List of Tables

3.1 The band describes the sizes of the filters used and the dimensions of the pooling window. There are two columns for filter sizes, the second column corresponds to the original version of HMAX. The third column corresponds to the sizes used in this thesis. For example, in the first band, the filters employed had a size of 7×7 and 9×9 in the original version of HMAX. However, in the version implemented for this thesis, the filter size for band 1 was only 7. The size of the pooling window is 4×4, 6×6, 9×9 and 12×12 for each band respectively. 44
3.2 Blob Detection Mechanism. The first column shows the steps in the BDM and the second column shows the image for the corresponding step in figure 3.8 and the MATLAB function employed for each step. 50
3.3 Performance comparison between the HMAX implementation of Serre et al using standard object libraries. The AI systems that they used in this comparison are a part-based generative model termed the constellation model (Fergus et al., 2003; Weber et al., 2000), a hierarchical SVM-based architecture (Heisele et al., 2002) and a system that uses fragments and AdaBoost (Leung, 2004). This table was extracted from (Serre et al., 2005a). 52
3.4 Performance (%) comparison using the COIL-100 library. The models were trained using 4, 8 or 18 training views and tested with the rest of the images in the image set (library), 6800, 6400 and 5400 testing views respectively. The results for the first three visual systems were obtained from (Roth et al., 2002). 55

5.1 Training scenarios for the embodied comparison of the models. The second column in the table shows the number of views that the models used and whether or not noise was added to the centroid of the blob detected by the BDM. 85
5.2 Number of correct guesses by the RBF and HMAX models for each scenario (out of 110 presentations during the test phase) when tested using trajectory 1. The RBF model performs better than the HMAX model in the four scenarios. 88
5.3 Sum of the normalised similarities for each region of the maps in figure 5.9. The values in this table were calculated by $\sum_{ij} sim^r_{ij}/max_r(sim)$, where sim_{ij} are the similarity values in the region r and $max_r(sim)$ is the maximum similarity value in region r. 91
5.4 Sum of the normalised similarities for each region of the maps in figure 5.10. . . . 92
5.5 Sum of the normalised similarities for each region of the maps in figure 5.11. . . . 93
5.6 Sum of the normalised similarities for each region of the maps in figure 5.12. . . . 94

6.1 Comparison of the performance (%) of the RBF model using SVP and DBCV using the four movement strategies during training and the test trajectory during the testing phase. The performance refers to the number of times the models guess correctly over the number of time steps in the test phase. 112

7.1 Unwrapping algorithm to convert panoramic images into landscape images. 129

7.2 RBF model performance. The left column shows the performance when it was evaluated in every position in the arena. The right column shows the performance when the model was evaluated only within the advantageous zone. 133

7.3 Performance of the RBF model when using the DBCV. The left column (general) shows the performance in every valid position in the arena and the right column (adv) shows the performance within the advantageous zone. 135

8.1 Reduction of the RBF model. See text for details. 141

8.2 Difference between the RBF_A similarity map and RBF_C similarity map. The values are the difference between the sums of the normalised values of each similarity map (for each object and trajectory). 145

8.3 Number of "bad" blobs in the movement strategies. Movement strategy T2 has the largest number of bad blobs amongst all the movement strategies. Object 6 is wrongly detected the largest number of times. 148

8.4 Total sum of the distances between the training views of the movement strategies for the RBF versions. For the three versions of RBF movement strategies T1 and T2 show lower distances compared with the distances for movement strategies T3 and T4. The distance values of the strategies for RBF_A are in bold and the largest distance values are in italics. 150

8.5 Thresholds for every object and every RBF version. 155

8.6 Total number of correct guesses by the RBF models when using the threshold based method to collect the training views. 155

8.7 Total number of correct classifications by the RBF models when using the neighbourhood based method to collect the training views. 157

A1 Number of correct guesses for each object when using the RBF_A, RBF_B and RBF_C models. The names of the variables denote the version of the RBF used, the movement strategy used to collect the training views, the value of sigma employed in the classifier, whether the blobs were corrected or not (in case the T1, T2, T3 or T4 were emploted) and the method to select training views in case the spiral movement strategy was used. For example, C-TS-σ1.8-NA denotes RBF_C (C), using the spiral movement strategy (TS) with $\sigma = 1.8$, employing the neighbourhood method to select the training views with the RBF_A (NA). B-TS-σ1.8-I3C denotes RBF_B, using the spiral movement strategy (TS) with $\sigma = 1.8$, the interval based view selection method with interval of 3 in clockwise direction (I3C). 169

Acknowledgements

I am extremely thankful to many people for helping me in so many ways and for making this an excellent experience. Firstly, I want to thank my first two supervisors Hilary Buxton and Emmet Spier for giving me the opportunity to start this journey and being so helpful in my first two years. I am also deeply thankful to my supervisors in the final years, Anil K. Seth and Andy Philippides for being extremely supportive, encouraging, and giving so generously when so much was needed.

I am grateful to the Mexican government, in particular to the National Council of Science and Technology in Mexico (CONACyT) for giving me the financial support to pursue my doctorate and live in England.

A very special mention to the glorious Friday Football (and his cousin Indoor Tuesday Football) for making my weeks so much fun especially to Patrick, John, Rob, Paul, Phil, Fabrice, Cristiano, Bart, Andy P, Andy F, for sharing most of the games and all the kicks, laughs, runs, and yes, sometimes some goals.

My thanks go as well to Rosario, Pablo, Damian, Ashish, Petros, Felipe, Omar, for sharing many fun times and talking to me about many other things apart from brains, neurons, vision, computers and robots.

Thanks as well to the CCNR members, especially to Rosario, Thomas, Eduardo, Peter, Marieke, and Jose for sharing office times, lunchtime, and more office times.

I would like to make particular note of Bart Baddeley, Linc Smith, Thomas Buhrmann, Eduardo Izquierdo and Neil Robinson for having useful conversations which contributed in many ways to this work. As well, thanks to anonymous reviewers for giving me suggestions to improve published papers in which some parts of this thesis are based on.

I also owe a special mention to Anil, Andy, Paul Graham, Andy Field and Jenna Bailey for being such a great help by editing and proof reading so much of my work.

Thanks to all my friends and cousins in Mexico for being so supportive in so many ways even though they probably do not know what my PhD is about.

A special mention to Jenna Bailey for sharing this part of my life and making it special.

Lovely thanks to my sisters Mayra and Ana for sending hugs, sweets and salsa, and music from Mexico.

I am deeply thankful to my brother Alfredo for being my best friend and being there when I needed him.

And my final and deepest thanks go to my parents, Rosi for supporting me and for having taught me so much, and Alfredo for being with me always.

Modelling Active Bio-Inspired Object Recognition in Autonomous Mobile Agents

Edgar Bermudez Contreras

Summary

Object recognition is arguably one of the main tasks carried out by the visual cortex. This task has been studied for decades and is one of the main topics being investigated in the computer vision field. While vertebrates perform this task with exceptional reliability and in very short amounts of time, the visual processes involved are still not completely understood. Considering the desirable properties of the visual systems in nature, many models have been proposed to not only match their performance in object recognition tasks, but also to study and understand the object recognition processes in the brain. One important point most of the classical models have failed to consider when modelling object recognition is the fact that all the visual systems in nature are active. Active object recognition opens different perspectives in contrast with the classical isolated way of modelling neural processes such as the exploitation of the body to aid the perceptual processes. Biologically inspired models are a good alternative to study embodied object recognition since animals are a working example that demonstrates that object recognition can be performed with great efficiency in an active manner.

In this thesis I study biologically inspired models for object recognition from an active perspective. I demonstrate that by considering the problem of object recognition from this perspective, the computational complexity present in some of the classical models of object recognition can be reduced. In particular, chapter 3 compares a simple V1-like model (RBF model) with a complex hierarchical model (HMAX model) under certain conditions which make the RBF model perform as the HMAX model when using a simple attentional mechanism. Additionally, I compare the RBF and HMAX model with some other visual systems using well-known object libraries. This comparison demonstrates that the performance of the implementations of the RBF and HMAX models employed in this thesis is similar to the performance of other state-of-the-art visual systems. In chapter 4, I study the role of sensors in the neural dynamics of controllers and the behaviour of simulated agents. I also show how to employ an Evolutionary Robotics approach to study autonomous mobile agents performing visually guided tasks. In addition, in chapter 5 I investigate whether the variation in the visual information, which is determined by simple movements of an agent, can impact the performance of the RBF and HMAX models. In chapter 6 I investigate the impact of several movement strategies in the recognition performance of the models. In particular I study the impact of the variation in visual information using different movement strategies to collect training views. In addition, I show that temporal information can be exploited to improve the object recognition performance using movement strategies. In chapter 7 experiments to study the exploitation of movement and temporal information are carried out in a real world scenario using a robot. These experiments validate the results obtained in simulations in the previous chapters. Finally, in chapter 8 I show that by exploiting regularities in the visual input imposed

by movement in the selection of training views, the complexity of the RBF model can be reduced in a real robot.

The approach of this work proposes to gradually increase the complexity of the processes involved in active object recognition, from studying the role of moving the focus of attention while comparing object recognition models in static tasks, to analysing the exploitation of an active approach in the selection of training views for a object recognition task in a real world robot.

Submitted for the degree of D. Phil.

University of Sussex

September, 2008

Preface

This thesis contains research work already presented in several peer-reviewed publications. Chapter 3 is based on the paper (Bermudez-Contreras et al., 2008), published in the Vision and Computing Journal. Chapter 4 is based on two papers presented at difference international conferences. The first part of the chapter is based on (Bermudez, 2007a) which was presented in the Students Meeting of the British Machine Vision Association (BMVA) (finalist in the BMVA Students Papers Competition) and on (Bermudez, 2007b) which was presented in the Genetic and Evolutionary Computation Conference (GECCO 2007). The second part of this chapter is based on (Bermudez and Seth, 2007) which was presented and published at the European Conference of Artificial Intelligence (ECAL 2007). Finally, parts of chapter 5 and 6 appeared in the Proceedings of the XI International Conference of Artificial Life (Bermudez et al., 2008).

Chapter 1

Introduction

When we open our eyes and the world is in front of us, many objects, shapes, and colours appear in this three dimensional environment. This is something that could seem trivial if we never stopped to carefully consider and analyse all the processes involved in this rich visual experience. It is when we stop to do so, that we realise the necessary complexity of the processes involved, and then it seems almost miraculous.

Figure 1.1: Brighton seafront on a Sunday morning.

The scene in figure 1.1 shows an example of how rich this visual experience can be. There are different people standing, walking and jogging, further back, sail boats, a pebble beach, and the sea in the background. This could be something that we see everyday and simply take for granted. However, the understanding of how we go from photons being reflected by shapes and regions and forming a two dimensional image on our retina, to being aware of people, stones, and the sea, seems significantly more complicated.

The processes involved in the description of the scene in the previous image, colour and shape detection, object recognition, depth information acquisition, etc., together, make up what is known as visual perception. Palmer (1999) defines visual perception as

> "... the process of acquiring knowledge about environmental objects and events by extracting information from the light they emit or reflect."

In this work I will focus on the object recognition aspect of visual perception, which is considered to be the ultimate goal of visual systems, due to its usefulness in the achievement of crucial tasks (Yantis, 2000). For example, object recognition has evolved in animals to aid their survival by identifying food and avoiding predators, but also to aid successful reproduction by identifying mates. Although some animals use multiple senses to perform these tasks, in others, like humans, vision plays a predominant role.

Understanding the processes involved in visual object recognition is a non-trivial task. Considering that vertebrates perform object recognition with exceptional reliability and in very short amounts of time, many models have been proposed to not only match their performance in object recognition tasks, but also to assist in the study and understanding of the visual processes in the brain. While there have been great advances in the object recognition models employed to perform different tasks, the visual processes involved in object recognition are still not completely understood.

In order to study object recognition in the brain, it is necessary to have an understanding of several disciplines such as neurophysiology and neuroanatomy in order to understand the neural processes at the single neuron level, neuropsychology, to understand what the different parts of the system do, and computation, in order to understand how all the processes are carried out. For the purposes of this work I do not present a detailed review of these disciplines (see (Booth and Rolls, 1998) for a detailed study of object recognition from these perspectives), although many results and ideas will be explained from different perspectives using terms and concepts from these fields.

One important point that most of the classical models have failed to consider when modelling object recognition is the fact that visual systems in nature are embodied and situated (Pfeifer and Scheier, 1999; Beer, 2003). Embodied and situated object recognition opens different perspectives in contrast with the classical isolated way of modelling neural processes, such as the active exploitation of the body to aid the perceptual processes, and the autonomous acquisition of the visual information. The fact that the visual information is actively acquired by the system and not presented to it by the experimenter has important consequences in the performance of the models (Bermudez-Contreras et al., 2008; Pinto et al., 2008). An important consequence of having control of the acquisition of visual information is the consequent selection of the visual information to be processed, and therefore, the possibility of exploiting regularities in the incoming visual information. Biologically inspired models are a good choice for the study of active object recognition, since animals are a working proof that object recognition can be performed with great efficiency as an active embodied system.

In this thesis I study biologically inspired models for object recognition from an active perspective. I demonstrate that by considering the problem of object recognition from this perspective, the computational complexity of hierarchical models of object recognition can be reduced. Additionally, I demonstrate that the exploitation of variations in scale and rotation in the visual information imposed by movement can play an important role in the performance of models of object recognition studied. Furthermore, I show that temporal information can be exploited to improve object recognition performance in visually guided tasks using movement strategies. In addition, I validate the results obtained from simulated experiments in the real world. Finally, I show that the complexity of the RBF model can be reduced to some degree when the regularities in the visual information are exploited, via movement, by actively selecting the training views.

The approach presented here proposes to gradually increase the complexity of the processes involved in active object recognition, from studying the role of moving the focus of attention while comparing object recognition models in static tasks, to analysing an active object recognition task in a real world robot.

1.1 Structure overview

Chapter 2 introduces the main concepts, approaches and methodologies used throughout this dissertation. Chapter 3 presents a deep study of two models of object recognition, the HMAX and RBF models. The first model is a biologically inspired model resembling the hierarchical structure of the ventral pathway in the visual cortex. In contrast, the RBF model is a V1-like model which represents objects using a combination of low-pass filters with different sizes and orientations. In this chapter the performance of the models is compared to some state-of-the-art models in a static general purpose object recognition task using the COIL-100 object library. Additionally, the role of a simple attentional mechanism is analysed for the RBF and HMAX models in static conditions. In chapter 4, a study of neuro-controllers for the exploitation of movement in simple visually guided embodied simulated agents is presented. Additionally, a methodology to use evolutionary techniques when simulating complex visual information is proposed. Chapter 5 and 6 investigate the exploitation of the variation in rotation and scale that simple movement strategies impose over the visual information in active visual systems that use the RBF and HMAX models. Chapter 7 validates the results and predictions from the simulation experiments by analysing the role of movement in the RBF model using a real robot. Chapter 8 analyses whether a reduced version of the RBF model can regain performance by exploiting regularities in the incoming visual information in the selection of training views using a real robot. In chapter 9, the contributions and limitations of this thesis are summarised. Finally, interesting avenues of research are proposed that would extend this work in several directions, followed by a general overview of the thesis.

Chapter 2

Active object recognition and autonomous mobile robots

In this chapter I introduce the main concepts that will be used throughout this thesis. First I will describe what should be understood as object recognition in this work and the main theories used to explain it. In the subsequent section, I will review some popular models for object recognition, both computer vision based models and biologically inspired models, particularly focussing on the HMAX and RBF models. Next, I will highlight important concepts to consider when studying visual processes in mobile agents and their implications for object recognition artificial systems. Finally, I will briefly describe the methodologies used in this thesis.

2.1 Object recognition

Object recognition is a very complex computational task that has been widely studied. Whereas visual systems in nature solve this task with exceptional reliability and speed, the performance of artificial visual systems is still far from their counterparts in nature, and the visual processes in the brain are far from being completely understood. Therefore, current research on object recognition serves two purposes. The first purpose is to extend our understanding of the visual processes in natural systems, which some researchers argue is a way to understand the brain in general (Ullman, 1996; Edelman, 1997). The second purpose is to build artificial systems that perform visually guided tasks, for example, navigation of vehicles on land, air or under the sea, in the supervision or assembly of manufactured parts in industry or in the analysis of microscopic or x-ray images (Arman and Aggarwal, 1993).

According to the literature, object recognition is generally understood as two distinctive tasks, identification (or labelling) and categorisation (or classification). Identification is understood as when, for example, I can identify “my shoe” rather than any shoe. In contrast, categorisation is understood as when, for example, I can determine if the object I am looking at is a “dog” rather than a “bird”. Although these tasks were originally proposed separately, they are recently being considered as two points in a continuum of generalization levels (Riesenhuber and Poggio, 2000; Palmeri and Gauthier, 2004). In this work I take this perspective and distinguish between the two only in the level of generalisation which the current object recognition problem needs.

Object recognition problems have generally, in the literature, been considered as supervised learning problems (Riesenhuber and Poggio, 2000; Palmer, 1999). A typical instantiation of this problem is to present an image to the system and its output is the label for any object in the image. There are two phases, during the first phase (training phase), the system is "trained" with a set of examples that are accurately labelled. The second phase (test phase) consists of presenting images to the trained system which labels objects present in the image.

Various object recognition models have been proposed with the purpose of understanding the visual processes in the brain or, in order to design artificial visual systems to solve different tasks. In this section I describe some of the main approaches to model object recognition. Both, for the understanding of neural processes or with the purpose of designing visual systems.

Most of the current object recognition theories assume that there are certain regularities within the object views that can be exploited by the recognition process. Therefore, most approaches for recognition can be classified based on how they deal with the variability across object views. For example, Ullman (1996) divides the approaches into invariant properties methods, parts decomposition methods and alignment methods. The first category assumes that certain simple properties remain invariant under object transformations (invariances, feature spaces, clustering, etc). The second category relies on the decomposition of objects into parts (structural decomposition, feature hierarchies, etc). The third class relies on the compensation between the viewed-object transformations and a stored template.

Booth and Rolls (1998) consider a more extensive classification of the models, namely, feature spaces, structural descriptions, template matching, invertible networks and feature hierarchies based on views. Their classification adds the last two classes to the classification of Ullman, considering invertible networks that can reconstruct their inputs and compare them with the presented view, and secondly, another class that includes models that use a hierarchy that starts with low-level descriptions and builds more complex features based on what was represented in lower layers (Booth and Rolls, 1998).

Another classification is proposed by Riesenhuber and Poggio (2000), who consider that object recognition models can be classified as view-based or object-centered categories. In the first class, objects are represented as a set of view-specific features so that the object recognition process is based on previously seen object views. In contrast, the models in the object-centered category "extract" structural features or parts of the object that are view-invariant in a 3D coordinate system centred on the object. The recognition system then matches the extracted parts with the stored structural descriptions of the objects (Marr and Nishihara, 1978; Biederman, 1987). One of the most important recent models using this approach is the recognition by components (RBC) approach. This approach uses primitive 3D structures called 'geons' to create descriptions of objects that are view-independent (Biederman, 1987; Palmeri and Gauthier, 2004; Riesenhuber and Poggio, 2000).

Finally, Wallis and Bulthoff (1999) summarise some of the current and popular recognition theories with four categories which include, extraction of 3D information, projective invariants, active shape matching and 2D-image-feature based. The first category proposes the extraction of 3D information from a scene which is then compared to 3D models (object based approach). The second category charaterises intrinsic elements of shape that are unaffected by projections onto surfaces. The third category proposes to match stored models with the stimulus using 3D or 2D transformations (anchor points). The fourth category proposes to extract features from multiple views to represent an object (feature-based multiple views). For a more detailed review of models and theories of perception see (Palmeri and Gauthier, 2004; Riesenhuber and Poggio, 2000; Peters, 2000; Palmer, 1999; Booth and Rolls, 1998). In this work I will employ the view-based approach to extract features to represent objects using multiple views for each object (view-based or feature multiple views based). In the next section I present a brief overview of some of the popular models of view-based object recognition employed currently.

2.2 Object recognition models

Most state-of-the-art object recognition models are view-based. There is neurophysiological and psychophysical evidence that supports that this type of model as more biologically plausible than object-based models (Logothetis et al., 1994, 1995; Booth and Rolls, 1998). Given that part of the motivation of this work is to study biologically inspired models, I only consider those that are view based. These view-based models are divided based on the way they extract the view-based features and how they represent objects. Although it is possible to find models that share features from both fields, computer-vision (machine vision) based models generally use statistical regularities extracted from the images, mainly using template or histogram systems (local patches, bag-of-features, nearest-neighbour, etc.) (Wang et al., 2006; Zhang et al., 2006a; Lazebnik et al., 2006), whereas biologically inspired models resemble, to some extent, the current understanding of the visual processes in animals (Riesenhuber and Poggio, 1999b; Poggio and Edelman, 1990; Serre et al., 2005b; Mutch and Lowe, 2006).

2.2.1 Computer vision approaches

In this section I briefly review some of the popular state-of-the-art computer vision methods for object recognition. This area is constantly growing, with pragmatic improvements and extensions of methods, showing excellent results in specialised object recognition related tasks.

A popular and successful computer vision model is the scale invariant feature transform (SIFT) which was proposed by Lowe (2004). This method transforms image data into scale-invariant local features points. It uses differences of Gaussians to detect points of interest across multiple scales of an image. It then selects the most stable points across locations and scales to different orientations. Using such local feature detectors for object recognition has proven to be a successful option in the computer vision community. Since it has been proposed, this method has been extended and evaluated under different conditions. For example, Mikolajczyk et al. (2005) compare several descriptors and detectors for categorisation tasks in which extended SIFT descriptors perform best.

Another popular computer vision method, called the local patch based approach (bag-of-features), has been recently used for class object recognition. This method is based on the use of extracted information at various regions in the image (local patches) which is used as building blocks of an image (Wang et al., 2006; Teynor et al., 2006). Lazebnik et al. (2006) proposed an extension of the bag-of-features method using an histogram based detection of local features to compute rough geometric correspondence at global scale.

Another method in this class is the nearest-neighbor approach which uses similarity measures to construct prototype category examples(Rosh, 1973). These similarity measures are based on different image features, for example, colour, shape, texture, etc. Recently, Zhang et al. (2006b) proposed an extension of the nearest-neighbour approach using support vector machines (SVM).

Most of the computer vision models use templates or histograms to extract features from the object views. Template-based models perform very well on object recognition of a single object category (e.g. faces, cars, etc.). However, these methods show limitations when the object is subject to appearance modifications, suffering from high specificity and therefore, lacking invariance to object transformations. Histogram-based models show a large amount of invariance to transformations but their performance drops for general object recognition tasks (i.e. with multiple object categories) (Serre et al., 2005b). Given that the purposes of this work is not only to study models of object recognition from an embodied and situated perspective, but also possibly gain some understanding of visual processes in animals, I will not concentrate on computer vision models. Rather, I will focus on biologically inspired models of object recognition, which are explained in the next section.

2.2.2 Biologically inspired approaches

Biologically inspired approaches consist of trying to understand or solve a problem by resembling organisms or their processes in nature. This approach is based on the observation that in many cases, organisms in nature are living proof of solutions for complicated problems. Object recognition in real time is one example of this. Following this type of approach has two beneficial aspects. On the one hand, it can serve the purpose of gaining better understanding of processes in nature. On the other hand, it can help to provide design ideas or solutions for complex problems.

Biologically inspired models of object recognition have been gaining interest because they perform very well for general purpose object recognition tasks (Pinto et al., 2008). The main purpose of these models is to gain or to extend the current understanding of the visual processes in animals. Furthermore, some researchers try to understand the brain by studying these visual processes. Therefore, this type of model is intentionally designed to resemble, to some extent, the processes in the visual system in animals.

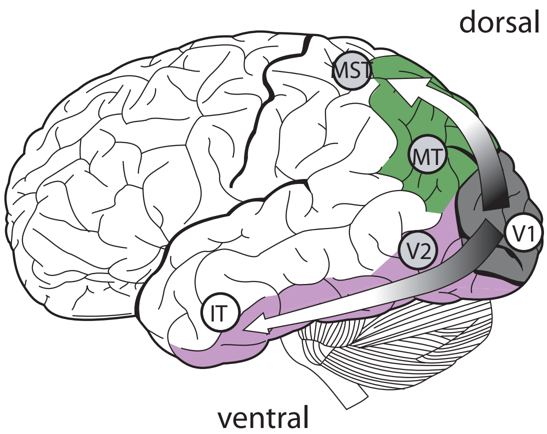

Figure 2.1: Ventral and dorsal pathways in the visual cortex. The activity of the ventral pathway is generally associated with the identification of objects while the dorsal pathway is commonly associated with the localisation and actions related to objects in space (image adapted from wikipedia).

The visual processes involved in object recognition in animals consist of high level vision computations which are associated with learning and classification tasks (related to object encoding schemes), as well as early vision processes where the features used to represent objects are extracted (Booth and Rolls, 1998; Wallis and Bulthoff, 1999).

In most of the current biologically inspired models for object recognition, objects are encoded as combinations or ensembles of simultaneously firing cells, in contrast with the early perspective of the 'grandmother neuron' in which every object is encoded as the response of a single neuron (see (Wallis and Bulthoff, 1999) for a review of object encoding). This is not meant to contradict the fact that there are cells in the brain with high specificity which respond to particular stimuli like faces (face-selective cells). Rather, instead of having only one cell responding to a particular face (grandmother cell), there is a set of face-selective cells which respond to a face. Furthermore, these cells do not respond to only one particular face but to a subset of faces (Booth and Rolls, 1998; Wallis and Bulthoff, 1999).

In general, biological inspired models can be divided in two types based on their use of extracted features. The first type of model matches views against simple templates. This type of model reflects the computationally economic visual systems of insects that actively align their body or eyes to match the current view to a template (Land and Nilsson, 2002; Cartwright and Collett, 1983). In contrast, hierarchical models try to reflect the current understanding of the visual system in primates. In this approach, a hierarchy is proposed where the complexity of the features increases through the levels of the hierarchical structure using combinations of features from the previous layers (Booth and Rolls, 1998; Riesenhuber and Poggio, 1999b, 2000). In particular, hierarchical models have been reported to be better than holistic single-template-based systems for static object recognition tasks (Serre et al., 2005b). Serre et al. (2005b,a) presented a modified hierarchical model based on (Riesenhuber and Poggio, 1999b) and reported it to be at least comparable to the best computer-vision based systems.

For the majority of this work, I am going to consider primate inspired and insect inspired models for object recognition. On the one hand, a hierarchical model that resembles the current knowledge about the ventral pathway in the visual cortex and, on the other hand, a simpler V1-like model which uses a template match based on simple features.

First, I describe hierarchical models that resemble the ventral pathway in the visual cortex in primates. Although there are many open questions about the neural processes underlying object recognition and categorisation in the visual cortex, there are some basic facts that are commonly accepted. Visual information comes from the retina to the primary visual cortex. Subsequently the processing of visual information is divided into two pathways: ventral and dorsal. The ventral pathway is associated with object identification and the dorsal pathway is associated with object localisation. The ventral pathway goes from the primary visual cortex V1 through extrastriate visual areas V2 and V4, to inferotemporal cortex, IT, and to prefrontal cortex (PFC), which plays an important role in linking perception to memory (Riesenhuber and Poggio, 2003). This pathway is a hierarchical structure with both the size of the receptive fields and the complexity of the detected features increasing as the information moves along the ventral pathway (Logothetis et al., 1994; Booth and Rolls, 1998).

There are several models that describe how this hierarchical structure in the ventral pathway can perform object recognition. These models are important because they have desirable properties for visual systems like scale and translation invariance. For example, Booth and Rolls (1998); Stringer and Rolls (2002) proposed a hierarchical model based on local inhibition and competition to describe feedback connections (and not only feedforward) to incorporate attentional processes in the model. For the purposes of this work, only feedforward models have been considered to explain low-level driven recognition.

The HMAX model

An important hierarchical model for object recognition is the HMAX model. It is a neuroscientifically inspired model, introduced by Riesenhuber and Poggio (1999b), that has recently gained attention by offering possible explanations for neuroscientific phenomena in the visual cortex (Deco and Lee, 2004; Riesenhuber and Poggio, 1999a). In its original version, the HMAX model describes a feed-forward hierarchical structure resembling the ventral pathway in the visual cortex (Riesenhuber and Poggio, 2000). Some modifications to this model have even incorporated attentional information by taking into consideration a winner-takes-all competition in the different levels of the model (Walther et al., 2002). (Serre et al., 2005a) reported that a version of the HMAX reflects physiological data and performs at the level of humans for restricted visual tasks not involving attention (Serre et al., 2005a). It is important to note that in this model the visual processing is driven by a bottom-up manner, the low-level information from the image drives the visual processes.

This model represents objects as the combined activation of a group of representatives of object classes. The activation of each of these groups is given by lower level patterns. Specifically, this model uses low-pass filters with different orientations over the image to be analysed and performs pooling operations over the output of these filters, having at the end, groups of representatives of object classes as shown in figure 2.2.

The different layers that describe the hierarchical nature of this model are S1, C1, S2

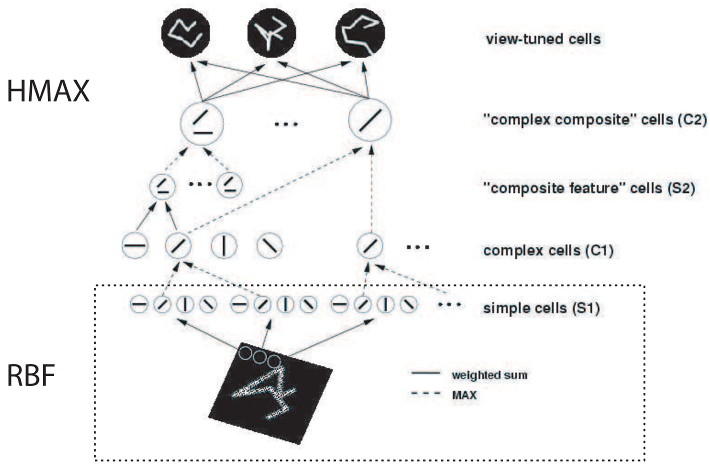

Figure 2.2: Hierarchical structure in the HMAX model. (adapted from (Riesenhuber and Poggio, 1999b))

and C2 (the details of the implementation will be described in the next chapter). The S1 (simple) layer applies low pass filters with orientation sensitivities and differing sizes over the image. Then, the output of the S1 layer is pooled in the C1 (complex) using the MAX operation over the different sizes and orientations. The MAX operation is defined as a nonlinear pooling operation over units (cells) in which the strongest afferent determines the output. After that, in the S2 layer, the output of the C1 layer is combined using different orientations of the filters in the windows and the Gaussian activation of the result is applied. Finally, in the C2 layer, the MAX operation is employed again over the output of the S2 layer. This will be described in detail in section 3.2.1.

In general, the HMAX model has desirable properties observed in visual systems. It shows a significant degree of translation and scale invariance (Logothetis et al., 1994) and it has shown a good performance compared with state-of-the-art computer vision systems (Serre et al., 2005b).

The RBF model

Another biologically inspired model which will be used throughout this thesis is referred to as the RBF model. This model is a template matching type model (more similar to the models used to explain snapshot matching in insects (Land and Nilsson, 2002; Cartwright and Collett, 1983)). The RBF model is a V1-like model in the sense that it consists of only the first layer of the HMAX model (see figure 2.2). This means that the output of this model consists of the application of low pass filters with multiple orientations and sizes to the incoming images. The performance of this type of model has proven to be at least as good as (if not better than) several state-of-the-art computer vision based models for particular conditions (Bermudez-Contreras et al., 2008; Pinto et al., 2008). In the rest of the thesis, this model and the conditions in which it exploits visual information will be extensively studied.

2.3 Active perception, embodiment, and situatedness

Having contextualised the models that will be studied throughout this thesis, it is important to discuss some of the conditions considered when studying biologically inspired models. Given that one of the main purposes of these models is to understand processes in animals, it seems natural to take into account the role of embodiment and situatedness as well as active perception. As brains are always embodied and situated, they autonomously interact with their environment using their bodies actively. Active perceptual processes are everywhere in animals, not only in visual perception, but in multiple sensory modalities which consists of movement of the sensory system or the body itself. For example, active echolocation in dolphins and bats, active touch during whisking in mice, active depth estimation in honey bees and praying mantis. In mammals, directable gaze is an important part of visual information acquisition (Land and Nilsson, 2002).

Even though taking into consideration the interaction between environment and body while modelling biological processes, such as object recognition, could seem obvious, historically, this has not been the case. Many models and theories of perception have considered visual processes as an analysis of 2D images where the visual information acquisition is static, influenced by the perspective of David Marr where he states that "vision is the process of discovering from images what is present in the world and where it is" (Marr and Nishihara, 1978). However, perception has also been considered as an active process (eg J.J. Gibson) where the active control of the sensory system determines the information to be processed (Bajcsy, 1988; Aloimonos, 1993; Ballard, 1991). Furthermore, it has been proven that the study of many neural models (including object recognition) have benefited by the consideration of these concepts (Pfeifer and Scheier, 1999; Aloimonos, 1993; Webb, 1996; Ballard, 1991). Below I describe these concepts and explain how they are employed through out this thesis.

2.3.1 Active vision and object recognition

In active vision, the control of acquisition of visual information is part of the system. It is well known that the restrictions imposed by the interaction between body and environment can facilitate visual processing. For example, problems that are ill-posed and nonlinear for static vision perspective, can become well-posed and linear from an active vision perspective (Aloimonos, 1993). There are several studies which use active visual processes. In artificial systems, for example, active vision has been mainly used for navigation, obstacle avoidance and perceptual discrimination. Nolfi and Marocco (2000) used evolutionary techniques to control robots that exploit proximity sensors actively to perform perceptual discrimination. Similarly, Harvey et al. (1994) evolved sensory and neural systems to discriminate triangles and squares in a gantry robot by actively sensing the arena. Suzuki (2007) used evolutionary techniques to find controllers that exploit body movements to study the development of receptive fields. In these experiments it was demonstrated that it was possible to exploit the interaction between body, brain and environment using an autonomously active approach. However, in all these experiments only very simple visual sensors were used. The visual systems employed were not complex enough to perform complex object recognition.

Several studies have examined active perception in more complex visual systems from an active perspective. For example, Gvozdjak and Li (1998), highlighted the importance of active vision in an agent for recognition tasks using a pyramidal template-based model. This work uses an object-based structural description of the objects. Andreasson and Duckett (2003) presented an exploratory study of object recognition using a mobile robot with an omni-directional camera. The robot tracks extracted low-level-feature points that are used to build histogram-based high level features for object identification. However, this work presents constraints on possible movements, orientation-dependent objects and hand pre-segmented images. Moreover, these methods which show promising results exploiting active vision in object recognition, do not consider biologically inspired models of object recognition.

Attention and recognition

An implication of taking into account the active control of the sensory system, is the active selection of the incoming visual information. This selection process is referred to as visual attention. In this work I consider that active vision includes both covert and overt forms of attention. The first one refers to, for example, moving our eyes towards a person in the room. The second one occurs when, for example, in a single retinal image we select a part of it without moving our eyes. See (Booth and Rolls, 1998; Palmer, 1999) to review this concept.

In this work, I explore attention as a mechanism to reduce the amount of visual information being processed by selecting parts of the visual field, rather than modelling mechanisms of attention *per se*. Additionally, attention can be regarded as a mechanism to provide translational invariance in the object recognition processes, as a consequence of an active vision approach. The need for such mechanisms for translational invariance is suggested by evidence from neurophysiology that raises doubts about our ability to perform position-independent object recognition (Kravitz et al., 2008). In particular it was shown that object recognition is significantly impaired in the parafovea and periphery (from as close as $2°$ from the original object). Therefore, this suggests that in order for an object to be encoded, the gaze has to be directed to within $2°$ from the object in a scene (Findlay and Gilchrist, 2003). In this work I employ an on-line blob detection mechanism (BDM) as a proxy for attention when studying object recognition processes (this mechanism is explained in detail in section 3.2.3).

2.3.2 Embodied and Situated visual systems

An important consideration when studying neural processes is the fact that they exist within a body which in turn is used by an agent to interact with its environment. Embodiment is a concept that stresses this fact. By considering the body as the medium through which an agent interacts with its environment, the regularities exploited by the sensory system, including the visual system, are determined by the characteristics of this dynamic interaction (for example movement). Additionally, the interaction between agent and environment is carried out autonomously, a concept referred to as situatedness (Pfeifer and

Scheier, 1999). A common baseline in the evaluation of the models for object recognition is that the visual information is presented to them and restricted by the experimenter, rather than the systems acquiring it by themselves. In some cases, these imposed restrictions can play an important role in the recognition process and hence, in the performance and evaluation of models (Pinto et al., 2008). On the other hand, a visually guided situated agent acquires the visual information solely on its own (without the experimenter completely determining which visual information is processed). One of the implications of autonomously acquiring the sensory information is the active control of the sensory systems. In this thesis, the concepts of embodiment and situatedness are not considered to be a discrete property (either present or not) but a continuum in the degree of embodiment and situatedness. For example, in chapter 3, the visual information is presented to the visual system as images and the visual system has a simple attentional mechanism (or blob detection mechanism) which determines which region in the image will be processed. This case would be less situated than the case of the simulated experiments in chapter 4, in which the visual information captured by a simulated video camera, which is attached to the body of autonomous agent, is determined by the autonomous control of its motors. Analogously, the restrictions imposed by the body of the agents, affect to different degrees the visual processing in the different experiments thought the thesis. The important point is to stress the the fact that the restrictions imposed by the way the visual information is presented to the system, have also to be considered in the analysis of the object recognition processes. This concept is analysed in chapters 7 and 8 in which the role of the features in the visual information determined by different movement strategies in the object recognition process is evaluated.

2.3.3 Movement and object recognition

Active vision considers the active selection of the incoming visual information. This selection process can be done by moving the sensory apparatus or moving the complete system. Insects, for example, perform characteristic eye, head or complete body movements to extract features to help recognition of depth, shape and size (Collett and Rees, 1997; Cartwright and Collett, 1983; Land and Nilsson, 2002; Bianco et al., 2000; Lehrer and Bianco, 2000).

In artificial systems, there have been relatively few studies where the exploitation of movement to aid object recognition processes has been considered. Some examples include, Bianco et al., proposed a model to exploit the vector fields produced by visual behaviors to explain how visual landmark learning works(Bianco et al., 2000). Borotschnig et al. (2000) use an active approach to use multiple views using the eigenspace approach (Murase and Nayar, 1995) to help the discrimination processes when a single view is not enough to unambiguously recognise an object. Another study, done by Arbel and Ferrie (2002, 2001) proposes a paradigm to facilitate object recognition in a supermarket checkout scenario when objects are presented to it by a human. In (Arbel and Ferrie, 2001), the authors propose an algorithm to recognise objects with a camera mounted on a mobile server. They use an entropy map to guide the camera along a trajectory to minimise the ambiguity in recognition. Finally, in (Roy et al., 2004) a review of active recognition approaches using view planning when a single view is not enough to recognise an object unambiguously is presented.

Movement has also been used in visual recognition, not only as a mechanism to extract or match features, but as a feature itself in the discrimination process. For example, in structure from motion (SFM), the structure extracted from 3D coordinates of moving objects is used for recognition. In contrast, motion based recognition approaches extract movement information from a sequence of images for the purpose of recognition. In (Cedras and Shah, 1995), the authors present a complete review of motion based recognition.

In this work I consider active object recognition as processes that extract features and regularities in the incoming visual information that are explited for object recognition and they are imposed by the control of the visual sensors (via body movement or attentional mechanisms in the visual system). These regularities and features are not only imposed in space, for example the object features that are exploited in visual stimuli but also in time, in the case of the exploitation of changes in the visual information imposed by movement.

2.3.4 Temporal information and object recognition

From an active vision perspective, mobile visual systems exploit visual regularities as they gather information by moving, not only in space but also in time. The role of temporal information has also been considered in visual recognition. For example, Watanabe et al. (1996) use optic flow analysis to perform object recognition in moving objects using their MOROFA model. Additionally, Chen and Chen (2004) proposed a framework for object recognition based on image sequences using the nearest feature line (NFL) method. This method is based on a collection of lines passing through every pair of points in the feature manifold used to represent the objects. Even when this work uses sequences of images to recognise objects, the changes of the objects in the image sequences are not necessarily produced by a particular motion of the visual system. In this thesis, I will exploit the extraction of visual features in sequences of images for visual object recognition. This concept will be analysed and expanded further in the chapters 6 and 7.

2.4 Controllers for autonomous visually guided mobile robots

In this thesis I will study object recognition for autonomous mobile agents. These agents acquire visual information by themselves (rather than through the experimenter) and exploit regularities by actively extracting features not only in space but also in time. The study of autonomous mobile robots has been an active research field with a wide range of applications, from robots exploring Mars, to robots cleaning floors. Like their counterparts in nature, mobile artificial agents need to control sensory and motor information in order to navigate and carry out tasks.

However, the design of artificial neural controllers for such machine systems is not an easy task. Over several decades, various approaches have been developed for the design and study of these kinds of neural controllers (Saffiotti, 1998; Bekey, 2005). Hand-designing is a very simple strategy to produce successful controllers for autonomous agents (e.g. Braitenberg vehicles). However, in some situations the sensory-motor interactions required in order to perform a particular task are not evident. In these situations, there are different approaches to find this type of controller. One such approach is Evolutionary robotics (ER), where simulated evolutionary processes are used to design controllers for autonomous agents. This approach has proven to be useful not only in the study and design of robot controllers (Harvey et al., 1994; Floreano et al., 2004; Nolfi and Floreano, 2002) but also in shedding some light on the understanding of cognitive phenomena (see (Beer, 1995, 2000; Harvey et al., 2005)) as well as in the exploration of vision morphologies (Cliff and Miller, 1996) and visual properties of sensors (Liese et al., 2001).

The Evolutionary Robotics (ER) approach uses evolutionary techniques (e.g. genetic algorithms, GAs) as a search algorithm to explore the parameter space of the controllers. The search of parameters is driven by a fitness function that selects the most successful controllers for the particular task to be studied. The controllers most commonly used in this approach are artificial neural networks (ANNs). This is a formalism inspired by computational model of cells in the brain. In this work I will use a commonly used model of ANN, a Continuous Time Recurrent Neural Network (CTRNN) (Beer, 1995; Funahashi and Nakamura, 1993; Beer, 2003).

There are two ways of carrying out experiments using this approach, using simulated agents and using real robots. In simulation the more important aspects are abstracted. The advantage of this version is that the evolutionary processes are not restricted to run in real time but can be faster. Additionally, the experimental conditions are easier to control and reproduce. However, the main disadvantage of this version is that finding a successful controller in simulation does not guarantee its success in a real robot. This is important if the processes we are interested in studying occur in the real world. On the other hand, using real robots allows us to study the phenomena we are interested in, in real world conditions. However, the disadvantages of this option is that sometimes the evolutionary process takes too long or the conditions for the experiments are difficult to reproduce or control.

In this work I will employ simulated visually guided agents in order to study biologically inspired models of object recognition and their exploitation of movement and temporal information. Furthermore, I will use a real robot implementation to validate the results and test predictions made using the simulation of visual processes.

Chapter 3

A first comparison: HMAX and RBF models in realistic conditions

3.1 Introduction

Object recognition is not a static process but rather, a continuous dynamic flow of information with variations in scale, illumination, rotation, occlusions, etc. These variations in the incoming visual information make object recognition a complex problem to solve for artificial visual systems. In contrast, visual systems in primates for example, perform object recognition tasks in complex environments with impressive robustness and reliability. Studying and modelling visual systems using inspiration from natural visual systems might be beneficial, not only to understand how neural object recognition processes work, but also as a way of providing useful insights in the way artificial visual systems are designed and evaluated. The HMAX model is a successful bio-inspired model proposed by Riesenhuber and Poggio (1999b) which resembles the hierarchical structure of the ventral pathway in the visual cortex. This model has been reported to have a certain degree of translation and scale invariance (Serre et al., 2005b,a). However, the computational complexity of the HMAX model makes it difficult to use in real time vision experiments. Natural systems use attentional mechanisms to reduce the amount of visual information to be processed (Itti and Koch, 2001; Walther et al., 2002). Here we therefore, study the role of a simple attentional mechanism by comparing a version of the HMAX model and the RBF model, a simple V1-like model, in an object recognition task.

In this chapter the two models for object recognition that will be studied through this thesis, the RBF and the HMAX models, will be presented. After that, these models will be compared to state-of-the-art visual systems and a validation of the version of the HMAX model used in this thesis will be presented. Once a performance baseline has been established for the versions of the object recognition models used in this thesis, an evaluation of the RBF and HMAX models incorporating an attentional mechanism for translation and scale invariance is presented. This evaluation shows that a simple V1-like model (RBF) can show comparable scale and translation invariances to the HMAX model when using an attentional mechanism under conditions that approximate a real world scenario in a mobile agent.

3.2 Visual system

Before explaining the details about the RBF and HMAX models, I will present the experimental setup in which these models are evaluated. The visual system implemented for the experiments in this chapter consists of two modules, the analysis module and the classifier module. The former is in charge of the analysis and processing of the incoming visual information, the latter is in charge of simulating the memory processes and the recognition output (see figure 3.1).

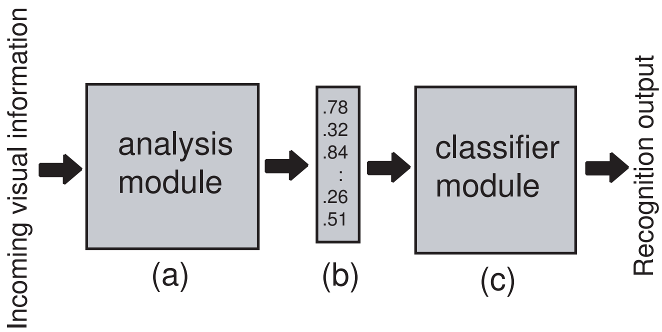

Figure 3.1: The visual system consists of the Analysis module (a) which can be either the HMAX or RBF model, and the Classifier module (c). The object recognition process starts with visual information coming into the visual system, then the Analysis module processes this information and outputs a vector (b) that the Classifier module analyses and characterises into the set of known objects. The output of the classifier module is the identifier (label) with the highest response for the input image.

3.2.1 The Analysis module

This module analyses the incoming visual information and outputs a vector that represents the object detected in the visual field. In this chapter, two models were implemented to process the incoming visual information, the RBF model and the HMAX model. The RBF uses a set of low pass filters to process the incoming visual information, providing a response similar to simple cells in primary visual cortex, V1 (see figure 3.2). In addition to processing the incoming visual information with low pass filters like the RBF model, the HMAX model has a more complex hierarchical resembling complex cells in the visual cortex (see figure 2.2).

The main difference between the two models is that the RBF model only filters the input visual information (similarly to the cells in V1) and uses this information to represent an object view. In contrast, the HMAX model has other layers that, by further processing, build a more abstract representation for each object view. The details about the two models are given below.

RBF model

This model consists of the application of low-pass filters with four different orientations and sizes over the incoming visual information, emulating the primary visual cortex, V1

(Howell and Buxton, 1995). The output of this model is a vector of all the outputs of the different filters applied over the input image. This process is illustrated in figure 3.2.

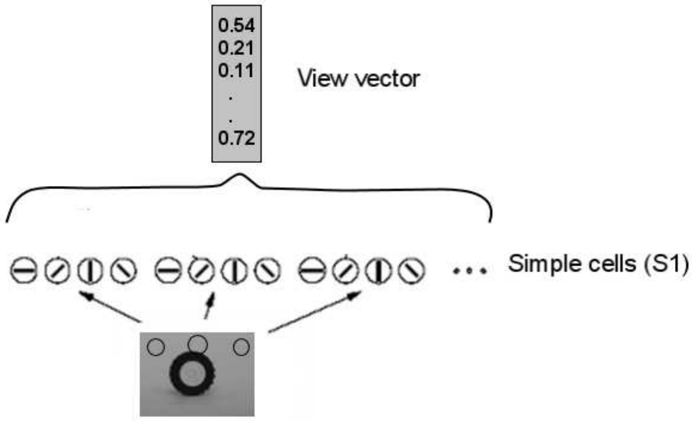

Figure 3.2: RBF model: The representation of an object view is given by the filtered incoming visual information using low-pass filters of different sizes and orientations. Three sampled regions are shown here.

The filters used are derivatives of Gaussians with four orientations (0, 45, 90 and 135 degrees). Every filter consists of a $n \times n$ matrix $F_{n \times n}$, where n can be $7, 11, 15, 21$. Every element f_{ij} in the filter F is calculated:

$$f_{i,j} = g(u, \sigma_u) \cdot g'(v, \sigma_v) \tag{3.1}$$

where $g(x, \sigma) = \frac{e^{(-x^2/2\sigma^2)}}{\sigma\sqrt{2\pi}}$ and g' is the derivative of g. The values of sigma $\sigma_u = \sigma_v$ were 1.75, 2.75, 4.25 and 5.75 for the different filter sizes, respectively. The values of u, v were worked out as follows:

$$\begin{pmatrix} u \\ v \end{pmatrix} = r * \begin{pmatrix} j - \frac{n+1}{2} \\ i - \frac{n+1}{2} \end{pmatrix}$$

and r is the rotation matrix

$$r = \begin{pmatrix} cos\theta & -sin\theta \\ sin\theta & cos\theta \end{pmatrix}$$

Finally, the filter F was normalised $F = \frac{F}{\|F\|}$ and $\| \cdot \|$ is the Euclidian norm. An example of these filters is shown in figure 3.3. Although different filters were tested (second derivative of Gaussians and Gabor filters), all the experiments reported in this thesis correspond to the filters in equation 3.1.[1].

The output of every unit in the RBF model ($s1_i$ cell) is the normalised filtered image

[1]The choice of this type of filter was based on the reported filters in the HMAX paper (Riesenhuber and Poggio, 1999b). However, later on, the authors of HMAX admitted to have used second derivatives instead. This mistake is reported in the Simple Filters section of the HMAX webpage `http://maxlab.neuro.georgetown.edu/hmax.html`

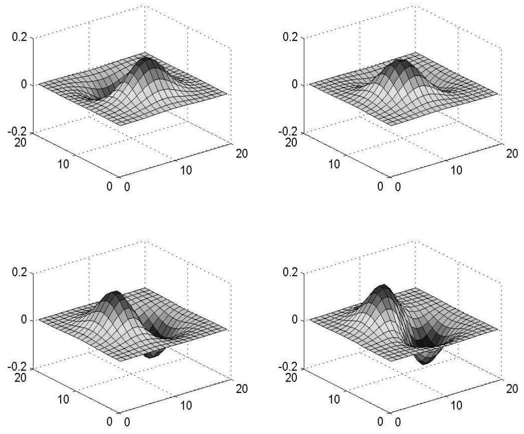

Figure 3.3: Example of filters with a particular size and four different orientations.

patch falling into its receptive field. That is, the S1 layer (or RBF model) is the collection of the convolved training views with the filters F with different orientations and filter sizes in equation 3.1.

$$s1_i = \frac{F_{o,s} \otimes I_i}{\sum I_i^2} \tag{3.2}$$

where $F_{o,s}$ is a filter as in equation 3.1 with orientation o and size s, $\otimes$ is the convolution operator and I_i the image patch falling into the receptive field of the $S1_i$ unit. The orientations and sizes of the filters determine the edges detected and the smoothness of the result (see figure 3.5 (a) and (b)). The S1 units are therefore sensitive to bars of different orientations, roughly emulating the response of simple cells in V1.

HMAX model

The HMAX model is inspired by physiological experiments in monkeys that suggest that object recognition in the cortex is mediated by the ventral visual pathway from primary visual cortex, V1, through extrastriate visual areas V2 and V4, to inferotemporal cortex, IT, where at each level of the hierarchy cells show an increase in the complexity of their preferred stimuli. Whereas V1 shows responses to small receptive fields and prefers bar-like stimuli, IT shows preference to more complex stimuli like faces with larger receptive fields (Riesenhuber and Poggio, 1999b, 2000). Based on these ideas, the HMAX model proposes a hierarchical structure composed of layers S1, C1, S2 and C2 (see figures 3.4, 2.2) . This hierarchical structure is similar to the structure found in the ventral pathway from V1 to IT. Each layer performs one of two operations: a weighted combination of simple features to build more complex ones or a maximum operation (MAX) in which the output of a unit is its strongest activated input unit. According to Riesenhuber and Poggio (1999b), the former operation increases the selectivity over the detected features by increasing their complexity through the different layers in the model. The MAX operation increases the translational invariance of the model by pooling over windows of the visual field with different sizes and locations.

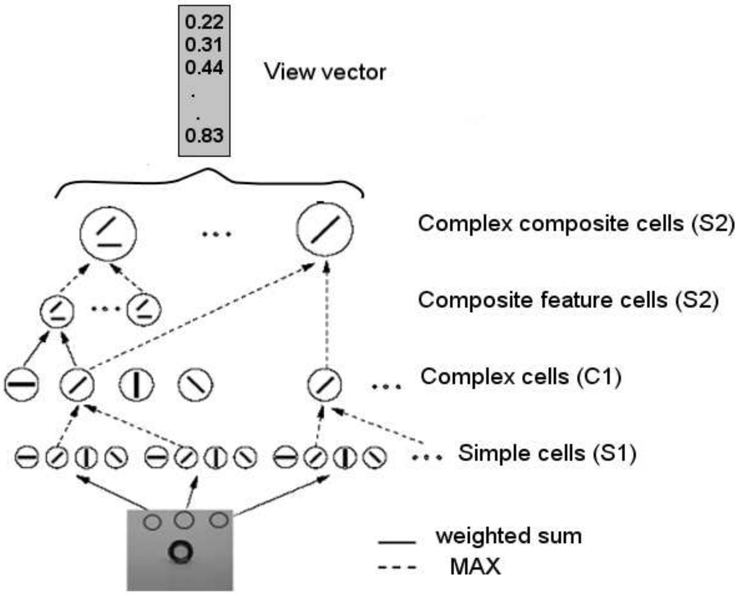

Figure 3.4: HMAX model: The representation of an object view is given by a more complex processing using a hierarchical structure. Figure adapted from (Riesenhuber and Poggio, 1999b)

The following description of the HMAX model and its implementation for this thesis are based on (Riesenhuber and Poggio, 1999b; Schneider and Riesenhuber, 2002; Serre et al., 2005b) and on the publicly available code in the HMAX web page[2]. Four different layers describe the hierarchical nature of this model S1, C1, S2 and C2.

S1 layer. This layer is in charge of convolving the image using filters with different sizes and four orientations (0, 45, 90 and 135 degrees). As mentioned previously, this layer corresponds to the RBF model. In the original version of HMAX (Riesenhuber and Poggio, 1999b), the sizes of the filters are organised in groups (or bands) of sizes (7×7 to 9×9 pixels; 11×11 to 15×15 pixels; 17×17 to 21×21 pixels; 23×23 to 29×29 pixels in two-pixel steps) described in table 3.1. However, for this thesis only one filter size is used for each band: 7×7 for the first band, 11×11 for the second, 15×15 for the third one and 21×21 for the fourth band (see column sizes* in table 3.1). This change in the implementation made it possible to have a faster implementation of HMAX. An analysis of these changes in the responses of the model and an evaluation of the impact in the performance of the HMAX model are presented in the following section.

The description of this layer corresponds to the description given for the RBF model. The output of the S1 layer is obtained as described in equation 3.2.

C1 layer. This layer takes the maximum activation of different groups of S1 units (pool) for different filter sizes with the same orientation (resembling the response of complex cells in striate cortex). The number of S1 units in each group (pooling range) which feed into each C1 unit is determined by the filter band (see table 3.1). For band 1, 4×4 S1 neighbouring units (for each size and orientation) are pooled into one C1 unit. For band

[2] `http://maxlab.neuro.georgetown.edu/hmax.html#code`

band	sizes	sizes*	pooling units
1	7, 9	7	4
2	11, 13, 15	11	6
3	17, 19, 21	15	9
4	23, 25, 27, 29	21	12

Table 3.1: The band describes the sizes of the filters used and the dimensions of the pooling window. There are two columns for filter sizes, the second column corresponds to the original version of HMAX. The third column corresponds to the sizes used in this thesis. For example, in the first band, the filters employed had a size of 7×7 and 9×9 in the original version of HMAX. However, in the version implemented for this thesis, the filter size for band 1 was only 7. The size of the pooling window is 4×4, 6×6, 9×9 and 12×12 for each band respectively.

2, S1 neighbouring units of 6×6 are pooled into one C1 unit. For band 3, S1 neighbouring units of 9×9 are pooled into one C1 unit and for band 4, S1 neighbouring units of 12×12 are pooled into one C1 unit. Only S1 units with the same orientation are pooled into a C1 unit to preserve feature specificity. The pooling operation that the C1 units use is the MAX operation which returns the strongest activation of the pooled S1 units. That is, a C1 unit $c1_i$ responds to stimuli with the same orientation of the S1 pooled units $s1_j$ that fed into it, but with the space and size invariance corresponding to the spatial and pooling range used for the respective band. In addition, C1 units are invariant to contrast reversal because before taking the MAX, the C1 units take the absolute value of the activity of the S1 input units. Each $c1_j$ unit response is given by

$$c1_i = max_j(|s1_j|) \tag{3.3}$$

where max selects the highest response over a pool of neighbouring $s1_j$ units. The activity of the C1 units is between 0 and 1. The receptive fields of C1 units have a 2-pixel overlap. That is, half of the S1 units in a pooling window were used also as input units of the adjacent C1 unit in each direction. Effectively, in the C1 layer the edges of the objects are highlighted (segments with higher pixel intensities). The MAX operation makes the detected borders expand. The thickness of the border depends on the number of pixels in the area considered when applying the MAX operation (number of units). This area can be imagined as a window shifting over the entire image. Every time this window is shifted, the maximum pixel value (brightest pixel) is stored in a C1 unit (see figure 3.5 (c)).

S2 layer. In this layer, a combination of the outputs of the previous layer is fed into S2 units. Within each filter band, a window of four adjacent non-overlapping C1 units is grouped as the input of the S2 units. The four possible positions of the C1 grouped units with four possible orientations give 4^4 possible configurations of units. The response of a S2 unit $s2_k$ has a Gaussian transfer function with mean 1 and standard deviation of 1.

$$s2_k = exp(-[\sum_{m}^{4}(c1_m - 1)^2/2]) \tag{3.4}$$

where $c1_m$ are the four C1 units in a one of the 256 configurations. The S2 layer provides the dictionary of features detected by the HMAX model, which is the combination of 2×2 arrangements of C1 units with four different orientations.

C2 layer. The highest layer of the model pools the MAX operation over the S2 units activation. This pooling operation gives the model size invariance over all filter size bands and position invariance over the whole visual field (resembling the response of cells in extrastriate area V4 or posterior IT). The result of this layer is a vector with 256 elements, each with the highest response for each type of S2 units (of the 4^4 combinations) at all positions and scales (see figure 3.5 (d)).

$$c2_n = max_m(s2_m) \tag{3.5}$$

Every $c2_n$ unit corresponds to the maximum activation of all $s2_m$ units of each of the 256 configurations. The C2 units provide the input to the view-tuned units (VTUs), which is where the learning occurs in the visual system (for both the RBF and HMAX models). The VTUs form the classifier system and will be explained in the following sections.

In general, the output of the analysis module is a representation of the object (a vector) being analysed in terms of certain features. The complexity and abstraction of these features is dependent on the model employed in this module. This representation (or view) of the object is characterised into the different classes of objects by the classifier module.

Differences in HMAX implementations. As previously mentioned, the implementation of the HMAX model for this thesis was based on the original version of the HMAX model (Riesenhuber and Poggio, 1999b) and the original implementation (HMAX_o) publicly available in the HMAX webpage[3]. Other implementations exist but were adapted for face recognition (Thomas Serre's implementation), or use different filtering parameters (Jim Mutch's implementation uses localised intermediate-level features), or increase the complexity and computational cost significantly (Minjoon Kouh's implementation uses more layers and soft-max operations). These versions of HMAX were not suitable for this thesis for different reasons. The implementation of HMAX used in this thesis (HMAX_m), although it is based on the original version of HMAX, differs from the original implementation of HMAX (HMAX_o) in two aspects:

- The number of filter per band. Due to computational complexity and time, in HMAX_m, one filter size was used per band (as opposed to several filter sizes per band in HMAX_o). See table 3.1.
- In HMAX_m the default filters used are first derivatives of Gaussians. In contrast, in HMAX_o the default filters are second derivatives of Gaussians. This difference was

[3]http://maxlab.neuro.georgetown.edu/hmax.html#code

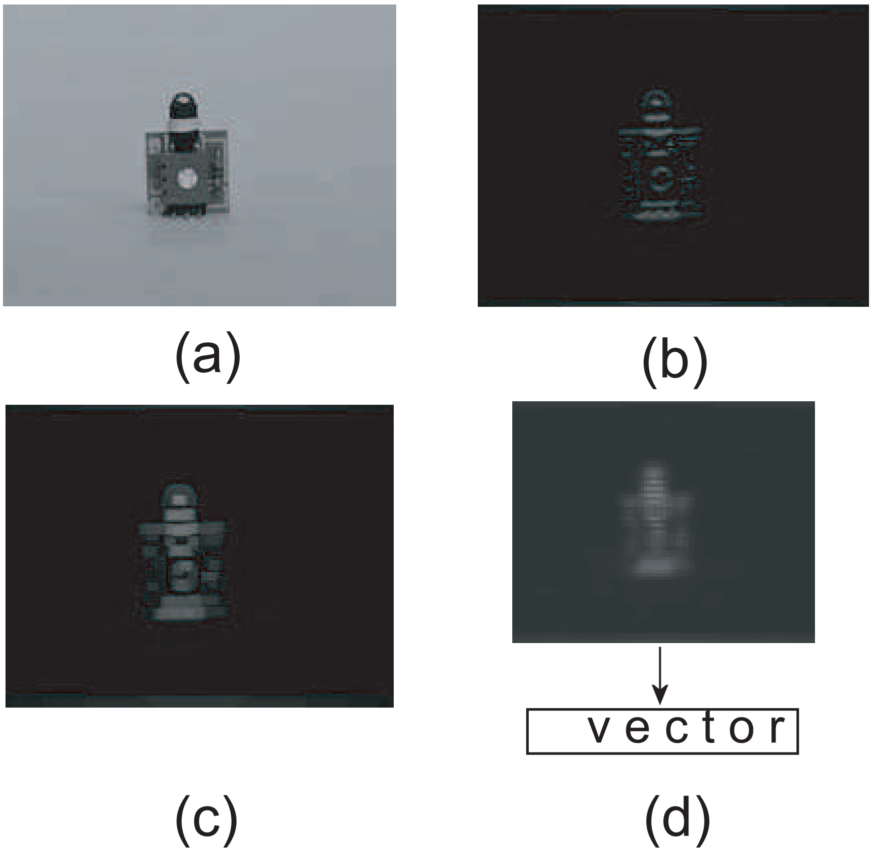

Figure 3.5: Illustration of HMAX layers shown in figure 3.4: (a) original picture. (b) image filtered after S1 layer, using a filter sensitive to horizontal segments. (c) image after layer C1, using the max pooling operation. (d) image after layer S2, applying the gaussian smoothing and subsampling. Finally, by taking the max operation over a combination of scales and sizes, a vector is obtained as a result of the max pooling operation.

due to an error in the report of the original paper (Riesenhuber and Poggio, 1999b) which is documented in the Simple Filters section of the HMAX webpage.

Here the main differences between the HMAX_o and HMAX_m are only outlined. An analysis of the differences in the outputs and performances of the implementations HMAX_o and HMAX_m will be presented and discussed in section 3.3.2.

3.2.2 Classifier module

The classifier module is based on the work of Edelman and Duvdevani-Bar (1997); Poggio and Edelman (1990). It uses view tuned units (VTU) to recognise objects. Each VTU is trained to respond according to the proximity (similarity) between the test view and the training views (see figure 3.12). That is, the more similar the test view to the training views, the stronger the response of the VTU. There is one VTU per object. Each VTU (see figure 3.6) is a set of radial basis functions (or RBF unit). A RBF unit is a Gaussian function G centered on each training view c_i for each object, that is, the centers were located at every training view. The response of each RBF is given by:

$$G(c_i, v) = e^{-\|c_i - v\|^2/\sigma_i^2} \tag{3.6}$$

where c_i is the centered-view vector and v is the vector that is being evaluated (test view).

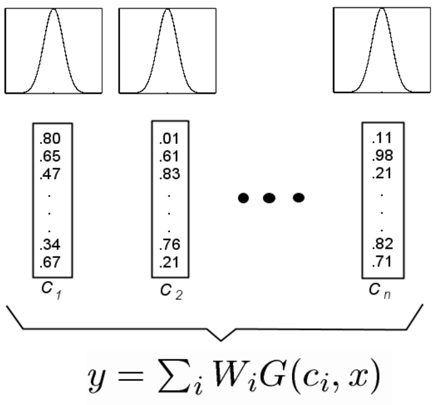

Figure 3.6: View Tuned Unit (VTU): each training view c_i is the centre of a Gaussian function. The more similar a vector x is to a centre, the stronger the response of the unit. The output of the VTU, $y = \sum_i W_i G(c_i, x)$

The response y of each VTU for a test vector x is given by

$$y = \sum_i G(v_i, x) W_i \tag{3.7}$$

where G is the activation of each Gaussian function centered on each view of each object. The response of the module y is the linear combination of weights W_i and the Gaussian G. The optimal weights W_i are computed in order to respond with higher values for similar views to views of the target object for a specific VTU and with lower values for the rest.

The procedure to work out the optimal weights is a well known optimisation method (described in (Orr, 1996; Rawlings et al., 2001)). In order to find the optimal weights W in

$$f(x) = \sum_i h_i(x) w_i \tag{3.8}$$

with training set $\{(x_i, \hat{y}_i)\}$, we use the least squares method to minimise the sum-squared-error $S = \sum(\hat{y}_i - f(x_i))^2$.

Therefore, in order to find the optimal weights that minimise S, we need to find the points where the derivative of S, with respect to the wights W, is zero. That is, we are looking for the values of W such that $0 = \sum_i (f(x_i) - \hat{y}_i) \frac{\partial f}{\partial w_j}(x_i)$, which is a number of linear equations with the same number or variables. And since $\frac{\partial f}{\partial w_j}(x_i) = h_j(x_i)$, then $\sum_i f(x_i) h_j(x_i) = \sum_i \hat{y}_i h_j(x_i)$. In matrix notation, $H^T F = H^T \hat{y}$. Since $F = HW$ (equation 3.8), then $H^T H W = H^T \hat{y}$, then $W = (H^T H)^{-1} H^T \hat{y}$. The solution is the so-called normal equation $\hat{w} = A^{-1} H^T \hat{y}$ where $A = H^T H$. Given that this is a well known method, just a brief explanation was given here, a detailed explanation of this method can be found in the appendix in (Orr, 1996).

For simplicity, the number of centers for each VTU was the same as the number of training views for each object. However, it is possible to optimise the classifier using different numbers of centres (Haykin, 1994). Given that one value for the width (sigma) of each Gaussian function was used for every center, sigma was selected initially as $\sigma = \frac{d_{max}}{\sqrt{2m}}$ where d_{max} is the maximum distance between the training views and m is the number of centers, which is the number of training views (Haykin, 1994). After, σ was optimised using a simple gradient descendant algorithm over a range of values to maximise the performance of the system around the initial point. This way of choosing the centers and width is considered "sensible" according to Haykin (1994), however, there are different methods of optimising the position of the centers and its width (Orr et al., 2000; Orr, 1998) which are more efficient. It is important to note that the goal of this work was not to achieve the optimal set of parameters but to get a sensible good performance.

This feature space described by the VTUs gives a parametric invariance (Edelman and Duvdevani-Bar, 1997), meaning that it employs different views from different perspectives (view point invariance) or illumination to recognise objects viewed from points with different illumination. This is sufficient in practice to support many different object recognition tests using similarity (Duvdevani-Bar et al., 1998). That is, using this multiple views approach, the visual system shows some degree of rotation and illumination invariance (Edelman and Duvdevani-Bar, 1997; Poggio and Edelman, 1990; Duvdevani-Bar et al., 1998) given by the parametric space generated by using multiple views to describe the object classes.

The visual system described so far presents the following limitations: 1) Difficulty in dealing with multiple objects in the visual field (some of them reported in (Schneider and

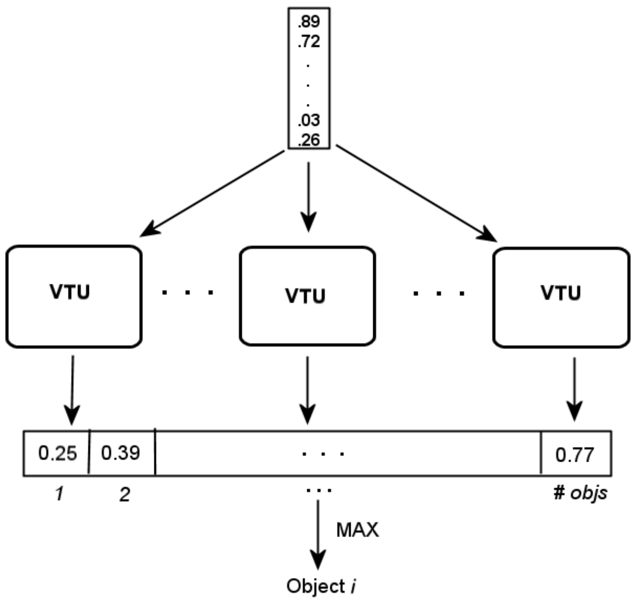

Figure 3.7: Classifier module: each object is represented by a View-Tuned unit (VTU). When a vector is analysed, the output of the classifier module is the maximum of all VTU responses.

Riesenhuber, 2004) for HMAX) and 2) not a large degree of translation and scale invariance when using the RBF module in the analysis module (less than 20 % for a translation range of more than 10 pixels, see figure 3.13(I) and figure 3.14(I)). In order to deal with these constraints, a simplified foveation mechanism was implemented.

3.2.3 The attentional and foveation mechanisms

Visual systems in nature are highly robust and reliable. An agent that is able to explore its environment requires a large degree of scale and translation invariance to perform object recognition tasks in realistic conditions. Given that an agent is able to move around its environment, the position and size of the object varies within the visual field. In many situations, the latency of response for such visual systems is crucial. Attentional mechanisms allow natural systems to reduce the amount of information to be processed. Similar requirements are needed for computer visual systems involved in object recognition tasks. Given the complexity and huge amount of visual information, in a natural scene for example, attentional and foveation mechanisms are usually employed for object recognition tasks in realistic situations (Itti and Koch, 2001; Paletta et al., 2005).

In the experiments for this chapter, a blob detection mechanism (BDM) was used to select or attend to a point in the visual field (Aloimonos, 1993; Bernardino and Santos-Victor, 2002). This attentional mechanism consisted of detecting blobs in the image using an edge detection function, then selecting a particular blob based on a certain criteria, and finally cropping and resizing the region with the selected blob. The criteria used to select regions in the image depended on the experiment to be carried out. In this chapter for example, the criteria used was to select the blob with the largest area.

The pseudo-code in table 3.2 shows the way the BDM was implemented for the experiments in this thesis.

Step	**Description**
1) Threshold the image and detect the edges using the canny edge detector	See figure 3.8B `graythresh()` `edge()`
2) Dilate edges using a gradient mask	See figure 3.8C `strel()` `imdilate()`
3) Fill the holes in the dilated regions	See figure 3.8D `imfill()`
4) Select the connected components (blobs) using a criteria (ie eccentricity, area, pixel intensity)	See figure 3.8E `bwlabel()`
5) Resize the selected blob	See figure 3.8F `imcrop()` `imresize()`

Table 3.2: Blob Detection Mechanism. The first column shows the steps in the BDM and the second column shows the image for the corresponding step in figure 3.8 and the MATLAB function employed for each step.

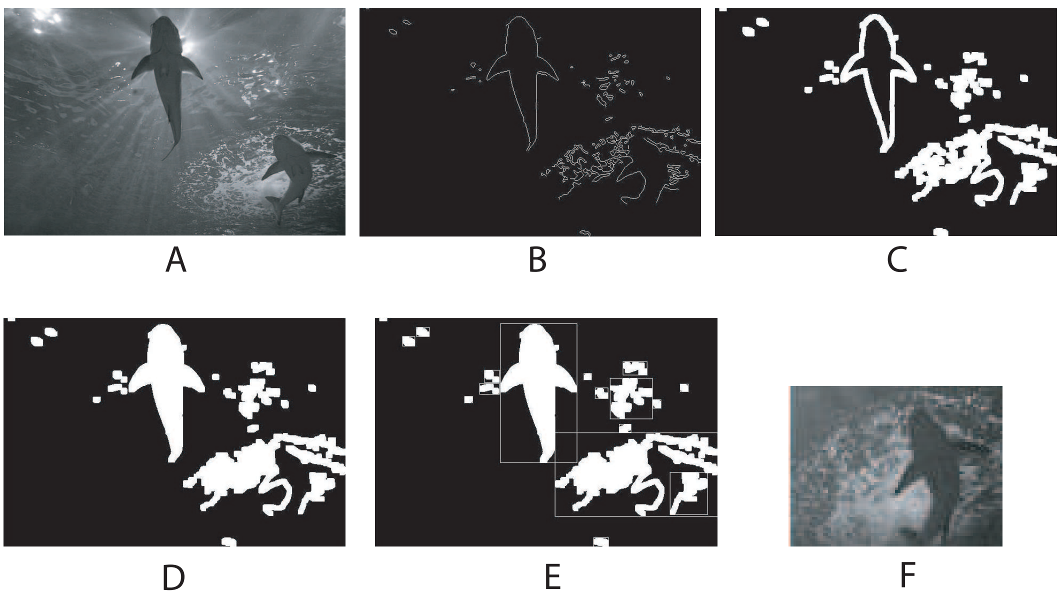

Figure 3.8: Blob detection mechanism. A) Image B) edges detected C) dilated edges D) Filled holes E) Connected components (detected blobs) F) Resized selected blob (in this case the largest area criteria was applied to select the blob).

The BDM was implemented in MATLAB. The `graythresh()` function was used to calculate the threshold of the input image. Then the canny edge detection function `edge()` was used to detect the edges. Following this, the edges where dilated in vertical and horizontal directions with the `strel()` and `imdilate()` functions. After that, the holes were filled with the `imfill()` function. The connected components (blobs) were obtained with the function `bwlabel()`. If more than one blob was detected, the blob with the largest area was selected. Other criteria were used in other experiments, for example calculating their area (chapter 3), pixel intensity (chapter 4), or eccentricity (chapter 7). Once a blob was selected, it was cropped from the original image using the `imcrop()` function, and finally it was resized to a standard size with the `imresize()` function.

The results reported in section 3.4 were obtained using a foveated area of 100×100 pixels (roughly the size of the largest object in the image database) and the visual field was 200x150 pixels.

3.3 Model evaluation

Once the RBF and HMAX models were described, a performance comparison of these models with state-of-the-art computer vision systems will be presented in this section. This comparison was carried out for object recognition task using a well-known image database. Furthermore, since the version of the HMAX model used in this thesis differs from the original version of HMAX (as mentioned at the end of section 3.2.1), a comparison of the outputs of these versions will be presented in this section.

3.3.1 State of the art comparison

The HMAX model is important not only because it allow us to study object recognition using a bio-inspired hierarchical structure of the visual cortex, but also because it shows useful properties like translation and scale invariance. The performance of the HMAX model has been compared to several state-of-the-art systems in the literature. For example, Serre et al. (2005a) has reported that a particular implementation of the HMAX outperforms several state of the art computer vision systems in several tasks using different conditions and databases (CalTech database: leaves, cars, faces, airplanes, motorcycles. Also they use the MIT-CBCL databases for faces and cars). The benchmark systems they used are a part-based generative model termed the constellation model (Fergus et al., 2003; Weber et al., 2000), a hierarchical SVM-based architecture (Heisele et al., 2002) and a system that uses fragments and AdaBoost (Leung, 2004). For details about the implementation of these systems and the comparison see (Serre et al., 2005a, 2004).

Table 3.3 shows that all the systems are significantly outperformed by the HMAX model (the details and references about the systems and datasets used in this comparison are in (Serre et al., 2005a)). This demonstrates that for standard object recognition tasks in specific controlled conditions, HMAX performs well. However, due to the complexity and time costs of the original version of the HMAX model, in this thesis a modified version of the HMAX model was used. In order to evaluate the differences between the original version of HMAX and the modified version used in this thesis, a comparison of

Dataset	AI system	HMAX
(CalTech) Leaves (Weber et al., 2000)	84.0	97.0
(CalTech) Cars (Fergus et al., 2003)	84.8	99.7
(CalTech) Faces (Fergus et al., 2003)	96.4	98.2
(CalTech) Airplanes (Fergus et al., 2003)	94.0	96.7
(CalTech) Motorcycles (Fergus et al., 2003)	95.0	98.0
(MIT-CBCL)Faces (Heisele et al., 2002)	90.4	95.9
(MIT-CBCL)Cars (Leung, 2004)	75.4	95.1

Table 3.3: Performance comparison between the HMAX implementation of Serre et al using standard object libraries. The AI systems that they used in this comparison are a part-based generative model termed the constellation model (Fergus et al., 2003; Weber et al., 2000), a hierarchical SVM-based architecture (Heisele et al., 2002) and a system that uses fragments and AdaBoost (Leung, 2004). This table was extracted from (Serre et al., 2005a).

the performance of the particular implementation of the HMAX and RBF models used in this thesis and the reported performance of several state of the art object recognition systems using a well-known standard object recognition library is presented. The library employed for this comparison is the Columbia Object Image Library (COIL-100) (Nene et al., 1996).

3.3.2 HMAX implementation validation

As mentioned previously, the two differences between $HMAX_m$ and $HMAX_o$ are the number of filters per band and the type of filter employed. The former difference is related to the computational speed in the original experiments carried out in this thesis. As mentioned previously, in $HMAX_o$, the number of filters sizes employed per band varied between 2 and 4. In contrast, in $HMAX_m$ only one filter size per band was used. The latter difference is the result of the reported mistake made by the authors of HMAX in the original paper. At the time the experiments of this chapter of the thesis were carried out, this error had not been reported.

The default type of filters in HMAX_o is second derivatives of Gaussians (Riesenhuber and Poggio, 1999b) but originally it was reported to have used first derivatives of Gaussians (this error is documented in the Simple Filters section of the HMAX webpage). In contrast, the HMAX_m uses first derivatives of Gaussians as default filters.

Figure 3.9 shows the output mean of the HMAX_m, HMAX_o and HMAX'_m implementations over 100 random 32×32 pixels COIL-100 images. The outputs of HMAX_m and HMAX_o are significantly different, which is not surprising given that the type of filters are different and also the amount of filters used per band is also reduced in HMAX_m (1 filter size per band) compared to HMAX_o. However, if we use the same type of filter in our HMAX implementation HMAX'_m and compare it to HMAX_o, 90% of the entries of the HMAX outputs are not significantly different (df=199, p¡0.05). Therefore, we conclude that the matlab implementation of the HMAX model is not significantly different to the original implementation when the same parameters are used.

Additionally, we can also see that the difference between the outputs of HMAX_o and HMAX_m is smaller when using the same images than when we compare different randomly

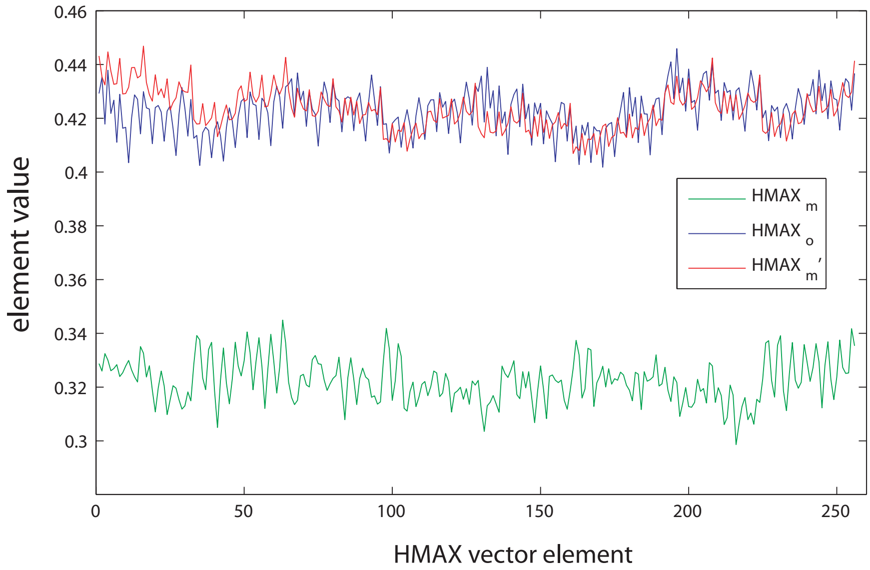

Figure 3.9: HMAX implementation output comparison. The lines represent the mean of the HMAX implementation outputs over 100 random 32×32 pixel black and white images from the COIL-100 library. The green line represents the output of HMAXm (one filter size per band and first derivative of Gaussian filters), the blue line represents the output of the original version of HMAX (HMAXo) and the red line represents the HMAX$_m$ implementation but using the same type of filter as in the original version.

selected images.

Figure 3.10 shows that the difference between the two implementations of HMAX (first column bar) is smaller than the difference between random pairs of different images being processed by the different implementations (second, third and fourth columns). The norm of the difference between random views after being processed by the HMAX implementations is compared. Namely, the outputs of HMAX$_o$ and HMAX$_m$ are compared using two sets I and J of 100 randomly chosen 32×32 grey-scale images each of the COIL-100 library.

Besides comparing the difference in their output, and after showing that if the same parameters are used, the original implementation is not significantly different to the implementation used in this thesis, we also evaluate how different in performance HMAX$_o$ and HMAX$_m$ are. We compared the performance of HMAX$_m$ and HMAX$_o$ with reported results for different object recognition systems using the COIL-100 library. The COIL-100 library has one view for every object at every 5 degrees of vertical rotation (72 views for each object). There are 100 objects (7200 views in total). Each view is a colour 128×128 pixels image. This library has been widely used to evaluate and compare object recognition models (Nene et al., 1996; Nayar et al., 1996; Chen and Chen, 2004; Roth et al., 2002).

In order to be able to compare the performance of the systems reported in (Roth et al., 2002) and the RBF and HMAX models, the images in the COIL-100 libreary were converted to 32×32 pixels grey-scale images (as reported in (Roth et al., 2002)). See some examples of object vies in this library in figure 3.11.

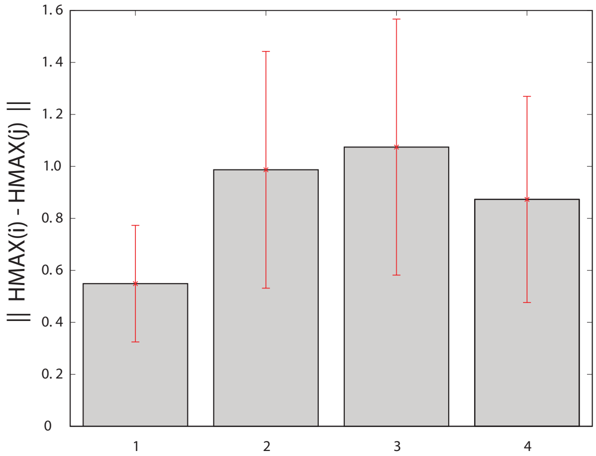

Figure 3.10: Difference in the output of HMAX versions. The first column is $||\mathrm{HMAX}_o(i) - \mathrm{HMAX}_m(i)||$, the second is $||\mathrm{HMAX}_o(i) - \mathrm{HMAX}_m(j)||$, the third one is $||\mathrm{HMAX}_m(i) - \mathrm{HMAX}_m(j)||$ and the fourth one is $||\mathrm{HMAX}_o(i) - \mathrm{HMAX}_o(j)||$, where $i \in I$ and $j \in J$ and $||\cdot||$ is the Euclidean norm.

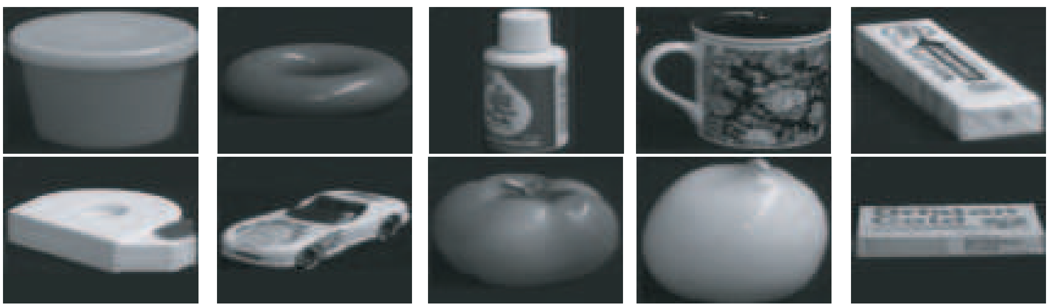

Figure 3.11: Coil library examples. Ten views of different objects of the COIL-library.

Table 3.4 shows the performance of several models reported in (Roth et al., 2002) using the COIL-100 library (first three rows). Additionally, this table shows the performance of the original version or HMAX_o(described in section 3.2.1), RBF_o which is the RBF model using the same number of filters as the HMAX_o implementation and, lastly, the RBF_m and HMAX_m which are the implementations of RBF and HMAX used in this thesis. The classifier was optimised in each model for the overall performance following the approximation method described in section 3.2.2. However, the RBF and HMAX parameters were kept the same as in the original description, even though the images' size for this comparison was only 32×32 pixels. We observe that, even though the outputs of the HMAX_m and HMAX_o are significantly different (when the same parameters are not used), their performance is not that different.

Model	4 views	8 views	18 views
SNoW w conjuntion of edges	88.28	89.23	94.13
Linear SVM	78.50	84.80	91.30
Nearest Neighbour	74.63	79.52	87.54
RBF_o	65.79	82.57	91.35
HMAX_o	60.05	76.09	88.31
RBF_m	67.41	81.82	86.66
HMAX_m	53.35	74.10	84.67

Table 3.4: Performance (%) comparison using the COIL-100 library. The models were trained using 4, 8 or 18 training views and tested with the rest of the images in the image set (library), 6800, 6400 and 5400 testing views respectively. The results for the first three visual systems were obtained from (Roth et al., 2002).

The performance of the RBF_m and HMAX_m is lower than the rest of the models. However, the performance of RBF_o and HMAX_o is closer to the performance of the rest of the models (particularly when using 18 training views). It could be possible that an optimised version of the models could improve their performance, for example by decreasing the size and width of the filters, or the size of the pooling windows. However, the purpose of this part of the thesis is not to find an optimal model performance for the COIL-100 library for these particular conditions, but only to have a rough estimation of the performance of the models in comparison with other standard systems using a well-known image library. This estimation shows that both models are suitable for general purpose object recognition under these conditions.

It is important to note the conditions in which this comparison was carried out. The objects in the COIL-100 database are centred with a uniform background. The rotation of the objects is uniform and controlled, similarly, the distance from the object to the camera is always the same in every image in the database, the images are histogram-stretched (see figure 3.11). These conditions are the same during testing. These restrictions are difficult to maintain in real vision, particularly in the case of a visually guided autonomous agent. The conditions in which the visual information is presented to the visual system are important because they can affect the performance of the visual systems in object recognition tasks (Pinto et al., 2008).

In this section we showed that the outputs of HMAX_m and HMAX_o implementations are significantly different. However, we also showed that when we use the same parameters, the output of the implementations are not significantly different. Additionally, we showed that their performance is not very different. Regardless these differences between HMAX_m and HMAX_o, it is important to note that the point of this chapter is to compare the performance of a simple V1-like model and a complex hierarchical model that was proposed to resemble to some extent the ventral pathway in the visual cortex (Riesenhuber and Poggio, 1999b) when a simple attentional mechanism is added to the models. Therefore, these differences are not relevant for the purpose of this chapter. Hereafter, we therefore use HMAX and RBF to mean HMAX_m and RBF_m respectively.

3.4 Comparison of the models in more realistic conditions

The experiments reported in (Riesenhuber and Poggio, 1999b) show that HMAX has a degree of invariance for 2-D transformations like translation and scale. The results in the previous sections show that the versions of the HMAX and RBF models show a decent performance when using multiple views and controlled conditions for general purpose object recognition tasks, which suit the initial requirement of a visual system for an autonomous visually guided agent. However, in contrast with the conditions of the previous experiments where the complete visual field corresponds to the object image, visual systems in nature deal with massive amounts of incoming information. In order to deal with this amount of visual information natural visual systems employ attentional mechanisms. Therefore, in this section I explore the impact of adding a simple attentional mechanism with the RBF and HMAX models.

There is evidence that suggests that not all the invariance in visual perception in natural visual systems is completely carried out by the visual processing itself. For example, there are mechanisms that facilitate the visual processing such as attentional mechanisms that reduce the amount of visual information that is processed. The advantage that an organism obtains from the cost of the additional developmental complexity required for a directable visual system is a reduced need for complexity in its neural circuitry and the concomitant energy requirements (Laughlin and Sejnowski, 2003). The analogous benefit of an active component to a vision system implemented on a serial processor is a reduction in computational load and, hence, an improvement in its latency of response. Although the HMAX model shows a significant degree of translation invariance (Logothetis et al., 1994; Riesenhuber and Poggio, 1999b, 2000), this property in a visual system can be provided by additional mechanisms that direct the visual processing into particular locations in the visual field (Aloimonos, 1993; Ballard, 1991; Bernardino and Santos-Victor, 2002). In this case, there is a possibility that the required complexity of the model, in order to perform object recognition, can be reduced.

3.4.1 Methods

In this experiment, a comparison of the translation and scale invariance is made between the RBF and HMAX models when aided by a simple attentional mechanism under certain conditions. Examples of training views of the objects are shown in figure 3.12.

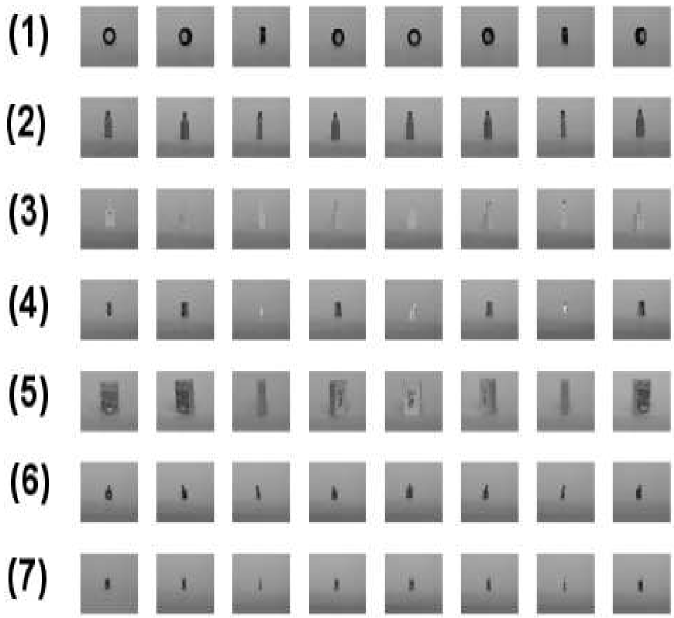

Figure 3.12: Image Set: object 1 (rubber wheel), object 2 (usb adaptor), object 3 (phone connector), object 4 (pencil sharpener), object 5 (deck of cards), object 6 (light sensor), object 7 (IR sensor). 8 views for each object.

The visual system described in section 3.2 was trained and tested in four different scenarios using both the RBF and HMAX models. The four training scenarios systematically increased the number of views, added the attentional mechanism, added natural illumination and changed the position of the objects. During the testing scenarios the translation and scale were varied systematically.

Training phase

Scenario I The objects were manually cropped and placed over a synthetic uniform greyscale background (similarly to experimental conditions in the original clip-like experiment (Riesenhuber and Poggio, 1999b)). Every pixel in the background has the same pixel intensity (200 from a scale 0-255). For this scenario no attentional mechanism was used. Instead, the analysis of the visual information was carried out over the entire visual field. Every object was represented using only one view (the first column in figure 3.12).

Scenario II The uniform background employed in scenario I was replaced with a nonuniform background with natural illumination as the visual field. The attentional mechanism was employed and every object was represented by only one view (again using the first column of figure 3.12). The amount of visual information was reduced by the BDM.

Scenario III Eight training views (8 columns in figure 3.12) and the attentional mechanism were used in this scenario. This means that not only was the background no longer uniform due to realistic illumination, but the multiple training views proportioned rotational variance to the visual system.

Scenario IV Eight views 8 and the BDM were used during training with natural illumination conditions as in scenario III. However, in this scenario the objects were placed in three different positions in the visual field before being processed by the BDM. In the previous scenarios, the object was always in the centre of the visual field (as figure 3.12 shows). The variation in the object location provided some degree of spatial invariance in the training view for this scenario.

The importance of these different scenarios is the fact that the local information around the objects in the visual field is increasing: in the first scenario there was no local information in the sense that all the pixels around the object had the same value. Then, the amount of information was increased in the second scenario using realistic illumination. In the third and fourth scenarios, the role of using multiple training views and the attentional mechanism is evaluated.

Testing phase

The conditions presented in the scenarios during the training phase were tested under the corresponding conditions described in the following scenarios.

Scenario I During testing, only one view per object is presented to the visual system over the same artificially uniform background as during training. For the translation experiments, the objects were shifted along the image by 1-pixel in each presentation. For the scale experiment, the image size was increased uniformly 1% every presentation.

Scenario II The artificially uniform background was replaced by a non-uniform background using natural lighting conditions. For the translation invariance experiment, the centre of foveation was shifted 1 pixel along the visual field in each presentation. For the scale invariance experiment, the size of the image before being processed by the BDM was increased 1% in each presentation.

Scenario III The background with natural illumination and the BDM were used in this scenario. During testing, a training view was randomly selected and modified according to the type of experiment. For translation, the centre of foveation was shifted 1 pixel in every presentation. For the scale invariance, the size of the image before being processed by the BDM was increased 1% in each presentation.

Scenario IV The testing of this scenario was the same as in scenario III. However, in this scenario the exploitation of local information was evaluated since during training the views had some degree of spatial variance.

The results of the translation and scale experiments described by the four scenarios is presented in the following sections.

3.4.2 Results

Translation Invariance

The results of the experiments in this section show how the performance of the HMAX and RBF models change under the conditions of the scenarios described previously. For scenario I, due to the fact that the artificial background was uniform and no attentional mechanism was used, the results of the MAX operation were the same everywhere in the visual field, allowing the performance of the HMAX to be high even after translating the object 20-30 pixels across the visual field. In contrast, given that the output of the filters in the RBF model contain spatial information (show clearly the edge, the shape of the object, etc), see figure 3.5b, this model is more sensitive to the position of the object in the visual field and its shape because the whole visual field is processed in this scenario. So, when the test views are translated in the visual field, the performance of the RBF model decreases considerably (see figure 3.13(I)).

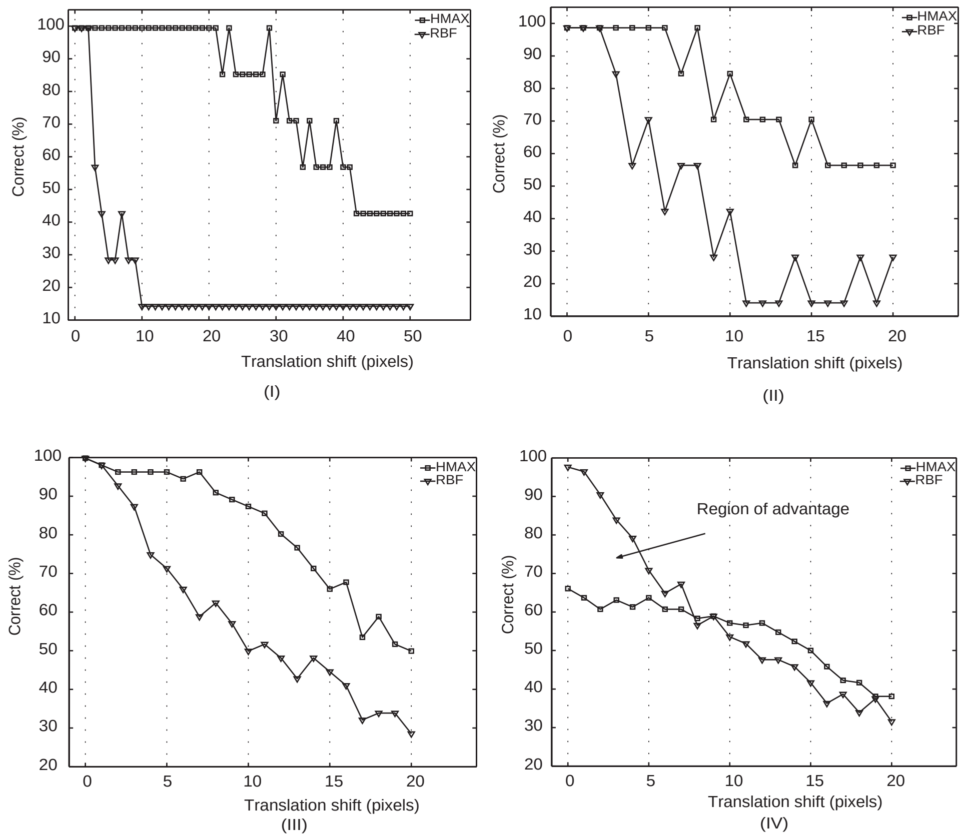

Figure 3.13: Translation invariance experiments. Scenario I: Uniform background, no foveation, 7 objs, 1 view. Scenario II: Non-uniform background, foveation, 7 objs, 1 view. Scenario III: Non-uniform background, foveation, 7 objs, 8 views. Scenario IV: Non-uniform background, foveation, 7 objs, 8 training views, testing in various positions, foveation.

For scenario II, we used natural lighting, a non-artificially uniform background and a single training view. The BDM (foveation) was used in this scenario instead of using the complete visual field as the input to the models (scenario I). When a non-uniform background (due to realistic illumination) was used, more local information was available. That is, the values of the pixels in the image around the objects were not the same in every position (in contrast with scenario I). As a consequence, the generalisation of the HMAX was not as effective as in the previous scenario. In contrast, as a consequence of attention and foveation mechanisms, the RBF model increased its robustness to translation in this scenario and its performance increased. See figure 3.13(II).

When more views were presented to the system in scenarios III and IV, more variance in illumination and pose were present in the training views (see figure 3.12). The variation in the local information (pixels surrounding the objects in the training views) increased with the employment of more training views. Again, this increase in the local information was exploited by the RBF model but decreased the performance of the HMAX model as shown in figure 3.13(III) corresponding to scenario III.

With more realistic conditions, where the object of interest can vary its position in the visual field, a system with an attentional mechanism can show performance benefits for object recognition tasks (Aloimonos, 1993; Floreano et al., 2004; Spier, 2004).

For scenario IV, the conditions during training were the same as in scenario III, however the objects were positioned in three different locations in the visual field during training in scenario IV. With this increase in the spatial variation in the training views, the performance of the HMAX model decreases significantly even for small shifts of the centre of foveation (see figure 3.13(IV)). In contrast, the RBF model is more robust to small shifts. This 'region of advantage' describes an interval where the foveation mechanism allows the RBF model to perform better than the HMAX model as long as the foveation centre is within a distance of 15% of the size of the object (the average size of the objects is 40 pixels and the largest foveation error in the 'region of advantage' is 6 which corresponds to 15% of the average size of the objects).

This experiment shows that when using a simple attentional mechanism, a simple model such as the RBF model can be more robust to translation shifts than a complex model such as the HMAX model certain conditions.

Scale Invariance

Another transformation in 2-D, where the HMAX model shows some degree of invariance according to Riesenhuber and Poggio (1999b), is the scale change of the objects analysed. We analysed the scale invariance of the two models described. The results found are similar to the translation experiments. In ideal conditions, HMAX performs better than the RBF model but when more realistic conditions are added, the RBF shows a better performance for a region of small perturbations.

Again, the experiments for scenario I, where the ideal conditions for the HMAX model show that when there is no noise (local information) on an uniform background, the HMAX model exploits the generalisation over scale to a moderate degree (10%). In contrast, the performance of the RBF model decreases faster, as is shown in figure 3.14(I).

When adding noise to the background (realistic illumination) and using attentional mechanisms in scenario II, the results show that HMAX decreases its performance while the RBF model keeps its performance very similar to the previous scenario, as is shown in figure 3.14(II).

In scenario III, when the diversity of the local information increases by adding more views, the performance of the models show that the RBF model can have a better performance than the HMAX model for very small values of scale change (see figure 3.14(III)). As before, the region described for these values where the performance of the RBF model is higher than the HMAX model is called the 'region of advantage'.

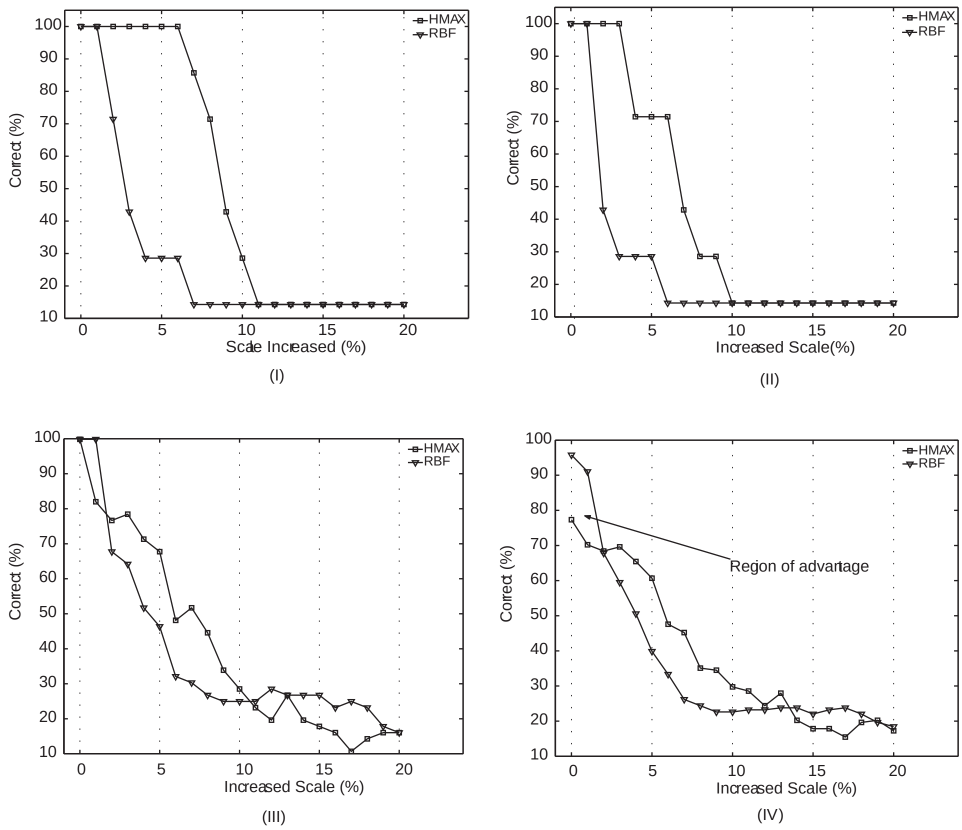

Figure 3.14: Scale invariance experiments: scenario I, scenario II, scenario III, scenario IV (previously described in figure 3.13).

Finally, in more realistic conditions, with different views and different illumination in scenario IV, the results show that using foveation as an attentional mechanism can improve the performance of the RBF model to some degree, similarly to the results for translation invariance. For this scenario, the 'region of advantage' goes from 0 to 2 % of scale, see figure 3.14(IV)). This illustrates the degree of scale invariance found when using low-pass filters and no other additional organisation in the model, suggesting that it is possible to reduce the computational complexity of the model and still have a degree of scale invariance using an attentional mechanism.

The explanation of the results of the translation invariance experiments also applies here but as the scale change represents more dramatic changes in the structure of the borders detected by the filters, the performance is affected more easily.

3.5 Conclusion

In this chapter, the performance of the RBF and HMAX models is compared in two sets of experiments. In the first set of experiments the performance of the HMAX and RBF models was compared to state-of-the-art computer vision models for general purpose object recognition using a well known object recognition database. Although it has been reported that the HMAX model outperformed most of the state-of-the-art models in single class object recognition tasks (Serre et al., 2005a), in this chapter it was demonstrated that the RBF and the version of HMAX implemented in this thesis have similar performances to three other visual systems when using the COIL-100 library. It is important to note that the HMAX and RBF models were not completely optimised for the particular characteristics of the COIL-100 images used in this experiment. However, there is room for a deeper optimisation of the models for this particular comparison using the COIL-100 library, for example, since the images were only 32×32 pixels, the filters could be optimised for small images, and the classifier could be optimised for the particular topology of the RBF and HMAX spaces with these characteristics. The reason for this was that the main purpose of carrying out this comparison was not to find the optimal parameters for the HMAX or RBF models when using the COIL-100 library, but to evaluate if the implementations of these models were comparable to other visual systems. Since one of the final aims of this work is to evaluate object recognition models for a mobile agent in a real world scenario, there is little value on concentrating on optimising the parameters of the models for this particular configuration. Additionally, in order to evaluate the difference between the original version of HMAX and the version employed in this thesis not only in performance, the outputs of these two versions of HMAX were compared using the COIL-100 library. It was shown that the implementation of the HMAX model used in this thesis is not significantly different to the original version of HMAX.

The second set of experiments in this chapter explored the translation and scale invariance of the RBF and HMAX models for object recognition tasks using an attentional mechanism. The conditions of these experiments were changed systematically according to four different scenarios. For conditions where no noise was present in the background (by using pre-segmented images) and where every object was presented using only one view (scenario I), the performance of the HMAX model was better than the RBF model. When an attentional mechanism was added, which simply cropped an area of the image around the object and resize it (scenario II), the performance of the HMAX model decreased with small perturbations in translation or scale (see figures 3.13II and 3.14II). In contrast, the performance of the RBF model was maintained for small perturbations when using the attentional mechanism, in comparison with the case when the whole visual field was considered. This is because within a subregion of the visual field, the invariance provided by the filters is enough for the RBF model to account for small perturbations. However, when the variance is increased during training by providing the models multiple training views (scenario III), the performance of the models decreases in a similar manner for small perturbations (see figures 3.13III and 3.14III). That is, a certain degree of robustness to perturbations is provided by using multiple training views. Finally, when the objects are positioned in different locations in the visual field during training (scenario IV), the performance of the RBF model was better than the HMAX model for small perturbations in the centre of foveation (translation) or small perturbations in scale.

To conclude, this chapter shows that in certain conditions, a simple model, like the RBF model, can maintain a degree of translation and scale invariance comparable to the ones in the HMAX model when using an attentional mechanism. This attentional mechanism also permitted a decrease in the latency of response of the visual system running on a computer. It is important to mention that even though the HMAX was not originally designed considering attentional mechanisms (Hung et al., 2005; Logothetis et al., 1994), vision systems in nature use these kinds of processes to deal with large amounts of visual information. Hence, it is important to consider these processes in biologically inspired models of the visual cortex. This consideration has proven to improve the performance of the HMAX (Walther et al., 2002), however, the role of the attentional mechanism in the performance of the model itself had remained unexplored (ie would the attentional mechanism improve the performance of a simple model as well?). In these experiments it was also shown by contrasting the two models analysed (HMAX and RBF), that when considering the attentional mechanisms and realistic conditions, the performance of a simple model can be as good as the performance of a hierarchical model like the HMAX.

Employing a simpler model such as the RBF model rather than the HMAX model can be useful for two reasons: 1) the visual information processing is less complex, and consequently has a shorter latency of response and 2) the position of the object, the shape of the object, details about texture, etc. are available in the RBF model, in contrast with the higher layers in HMAX where information is lost. Although the loss of this local information in the higher levels of HMAX allows it to have a larger degree of translation and scale invariance, the presence of local information in the RBF model could be useful to discriminate objects in complex scenes (Deco and Lee, 2004; Lee, 2003; Zhaoping, 2002).

Chapter 4

Simulated Embodied Visual Systems

4.1 Introduction

In the previous chapter I demonstrated that a simple V1-like model can maintain a significant degree of translation and scale invariance when using an attentional mechanism in realistic conditions. This therefore provides an example that an active vision strategy can reduce computational processing by exploiting attentional mechanisms.

In order for such a simple model to perform object recognition with the desired levels of translation and scale invariances in an autonomous mobile agent, it is necessary that the controller of such agent can provide the required movement strategies to perform the visual processing. The basic movement strategies required for this task are for example, to approach the object, and to maintain the object in the visual field. In this chapter I therefore explore the properties of neuro-controllers for autonomous mobile agents that can provide these required movement strategies.

Visual systems in animals evolved from primitive light detectors into directional and spatial light sensors when motility represented a great advantage with the appearance of predators (Fernald, 2004; Land and Fernald, 1992; Land and Nilsson, 2002). In this experiment, this evolutionary process in visual sensors in nature is mimicked.

In the first experiment of this chapter, controllers are evolved to perform phototaxis using panoramic or directional sensors. In this experiment it is shown that by restricting the sensory system, a simplification not only of the interaction between the environment and the agent, but also of the neural controller, can be achieved. The behavioural result of this restriction in the visual sensors improves the performance of the agents for this task by producing more reactive agents. Even when the visual system used by the agents is very simple, the results of this chapter stress the relation between sensors, controllers and motors from an active perception perspective.

In the second experiment possibility of using an evolutionary approach to find controllers for autonomous agents using simulated video cameras is explored. However, since the employment of that type of visual system imposed the simulation to run in real time and such a restriction makes the ER approach not suitable (Jakobi, 1998), a solution is proposed to overcome this problem. In this solution, I construct a simplified simulations of simulations. I illustrate the value of this by evolving controllers to perform object approaching and object discrimination using the simple simulation and show that such controllers transfer successfully into the rich visual simulation, despite significant differences in the structure of their sensory input.

4.2 Experiment 1: Using an active approach in a simple simulated agent.

This experiment compares the neural controllers and behaviour of artificially evolved agents using panoramic and directional sensors. I first introduce the methodology used in this chapter, I then describe the task and the experimental set up. Following this, I analyse the neural dynamics of the successful controllers and show that the use of directional sensors makes the required neural controller less complex than when panoramic sensors were employed. Finally, I conclude Experiment 1 with a discussion about how the sensors used affect the dynamics of the neural controller but also in the agent's behaviour.

4.2.1 Methods

The first experiment was carried out in an unlimited simulated arena. An object (target) was placed in the centre of the arena. This object emits a signal uniformly dispersed in the arena. For the purposes of this work, we assume that this signal is light. At the beginning of each trial, an agent was placed in a random position and random orientation within an area of 10×10 units around the object. Agents were evolved to perform phototaxis, that is, to approach the light source (object) guided by the intensity of the light emitted by the object. During the evolutionary phase, the agents fitness was evaluated as average performance over 5 trials of 300 time steps each. During the testing phase, the best evolved controllers were analysed over trials of 800 time steps.

The agent

The agent has a circular body with radius of 0.5 units and two wheels on each side driven by independent motors. The agent is able to sense the signal emitted by the object through two sensors placed at $\pm\pi/4$ radians from the line of orientation of the body (see figure 4.1). The body of the agent was symmetrical with respect to the axis of orientation.

A primitive visual system

The two sensors in the agents can be panoramic, or directional. In the first case, the sensors perceive light coming from any direction (even from behind the agent). In the second case, the sensors only perceive light coming from a particular region in front of the agent (see figure 4.1). For both cases, the activation of the sensors S is the inverse of the distance d between sensor and object:

$$S = \frac{1}{d} \tag{4.1}$$

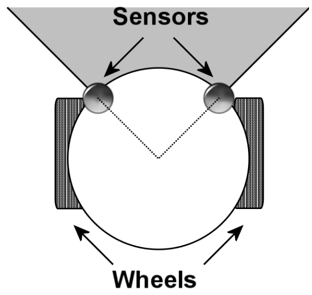

Figure 4.1: Agent body: two wheels on each side driven by independent motors. Two sensors placed at $\pm\pi/4$ radians from the line of orientation of the body. For directional sensors the agent can only perceive light coming from objects in the grey area. For panoramic sensors the light can be perceived from any direction.

Controller

The controllers for the agent are Continuous Time Recurrent Neural Networks (CTRNN). These kinds of artificial neural networks show desirable properties as robot controllers. A CTRNN shows complex dynamics and is a universal approximator (any smooth dynamical system can be approximated by a CTRNN with any degree of accuracy) (Funahashi and Nakamura, 1993). See (Beer, 1995, 2003) for a discussion and examples of how to use this type of controller to analyse cognitive phenomena using dynamical systems.

The state y of neuron i changes in time according to the differential equation:

$$\tau_i \frac{dy_i}{dt} = -y_i \sum_j w_{ji}\phi(y_j + \beta_j) + g \cdot I_i \tag{4.2}$$

That is, the state of each neuron is the integration of the weighted sum of all incoming connections (plus a gained input $g \cdot I$ for input neurons, where $g = 2$). ϕ is the sigmoid activation function, τ_i is a time constant, β_j is a bias, and w_{ji} represent connection weights from neuron j to neuron i. Parameter values for all neurons were randomly uniformly initialised in the following ranges: $\tau \in [0.2, 2.0]$, $\beta \in [-10, 10]$, and connection weights $w_{ij} \in [-5, 5]$. Solutions of y was calculated by numerical integration using the Euler method with time step $dt = 0.01$.

Initially, the controller consisted of eight neurons, specifically, two sensor neurons, four fully connected interneurons and two motor neurons. Another set of experiments was carried out using a neural controller with six neurons, two sensor neurons, two interneurons and two motor neurons (see figure 4.2).

The sensor neurons were activated by light (sensed as the inverse of the distance between each sensor in the agent and the object). The motor neurons received activation from every interneuron only. The output of the motor neurons was connected to the motors of the wheels of the agent with a gain of 2. Due to the nature of the body of the agent, a bilateral symmetry was imposed on the neural controller of the agent. Due to the symmetry of the controller, if the activation is the same in all neurons, the agent is going to go in a straight line indefinitely. However, any difference in the sensors breaks the symmetry due to the mutual inhibitory connection in the interneurons. So this repellor

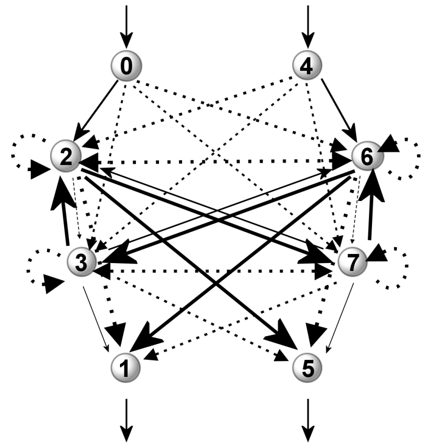

Figure 4.2: Neural controller: a CTRNN with 8 nodes. Neurons 0 and 4 are the sensor nodes, neurons 2, 3, 6 and 7 are fully connected interneurons and neurons 1 and 5 are the motor neurons. The width of each arrow represents the strength of the connection (weight). The solid lines in the arrows represent excitatory connections and the dotted lines in the arrows represent inhibitory connections.

occurs only when the line of direction of the agent intersects with the center of the object (being exactly in the middle, in front or behind). In that case, the activation of both sensors is exactly the same. In order to avoid this situation in the controller, random noise was added to the sensors.

Genetic Algorithm

A standard distributed mutation-only GA (using only mutation, no crossover) was employed to evolve the neural controllers for a phototactic task. A population of 400 individuals was evolved with mutation probability of 80% and 20% for mutation for each component and an elitism probability of 80%. The genome of each individual was coded in a real vector of 25 elements, 4 for the time constants of each neuron, 4 for the bias of each neuron, 1 for the sensor gain and 16 for the weights. Each element was coded as a real number in $[0, 1]$ and linearly scaled according to the parameters described in section 4.2.1. The fitness function F was defined as $F = 1/d_f$ where d_f is the distance from the agent to the object at the end of the trial. The final fitness of each individual is the averaged F over 5 independent trials. In this way the evolutionary pressure was towards individuals finishing as close as possible to the light source.

4.2.2 Results

The controllers were evolved to perform phototaxis in order to explore the role of different types of light sensors and different configurations in the neural controllers.[1]

Panoramic light sensors

After several thousands of generations, the evolved agents performed phototaxis successfully using panoramic sensors and 8-neuron controllers. It is important to mention that

[1]Neural controllers were also evolved using light sensed as $1/d^2$ under the same scenarios showing qualitatively the same results as the ones presented in this section.

controllers with less neurons were also initially used (5 and 6 neurons) but it was not possible to find successful controllers using such setup in a reasonable amount of experimental time (tried up to 20 thousand). Of course, this does not meant it is impossible to evolve controllers with panoramic sensors using less than eight neurons for phototaxic behaviour but rather, that it is difficult to evolve them, suggesting that more neural resources could be needed.

Most of the successful agents approached the object in a straight line and then remained close and continuously circled or "patrolled" it . This bevaviour can be understood by examining the internal dynamics of the best evolved controllers (see figure 4.3).

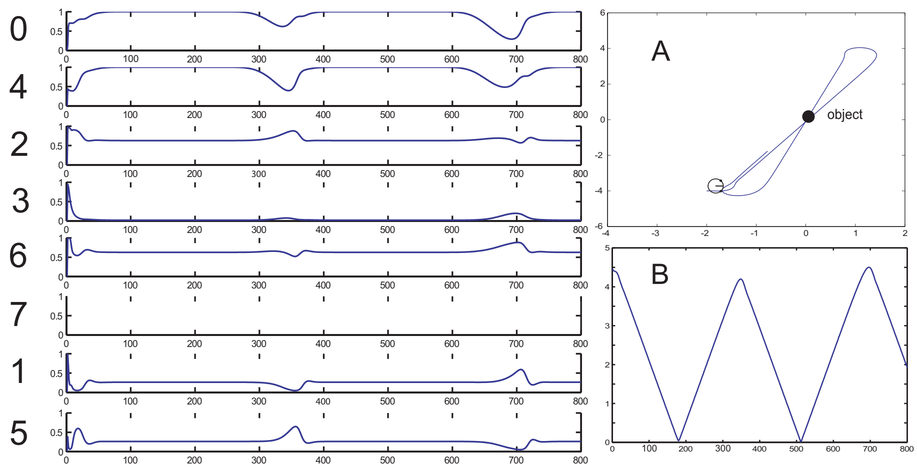

Figure 4.3: Dynamics of an evolved controller of 8 neurons using panoramic sensors (left panel). Neurons 0 and 4 are sensor neurons, neurons 1 and 5 are motor neurons and neurons 2, 3, 6 and 7 are interneurons. (A) Positions of an evolved agent during a test run. The object (target) is placed in the center of the arena (0,0). (B) Distance between the agent and the object during the test run (timestep vs distance).

Around timestep 190, the agent gets very close to the object (figure 4.3B). However, the sensor neurons were saturated well before this and remained so even after the agent passed the object. If we observe what happens after timestep 190, when the agent passes the object (so the distance increases), we can see that just before timestep 300 (see the output of the neurons 1 and 5 in figure 4.3A) the agent starts to change direction and returns to the object. This time corresponds to the time when the sensor neurons start to deactivate again (so the searching behaviour is once again triggered).

How can we explain the behaviour of the agent? How are these decisions made? A full dynamical systems analysis of the 8 differential equations that describe the system seems too complex to be able to explain the behaviour of the agent in general. Also, it seems too difficult to analyse the network structure of the controller (figure 4.2) as a way to explain the interactions and roles of the neurons. However, we can understand the evolved controller if we analyse all the possible states the agent can be in using a steady-state map.

We generate this map by placing the agent in a fixed position and record the neural activity after the network has stabilised (after approximately 50 time steps). Then we change the position of the object and record the neural activity of the stabilised network

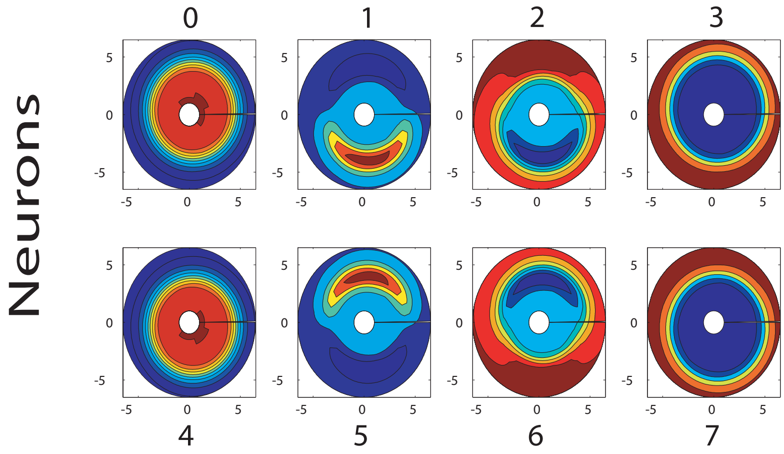

Figure 4.4: Long term steady state of the neural controller: the agent was fixed in a position facing right (indicated by a line) and the object was moved around it. After 50 timesteps the activation of each neuron is stored. Red regions represent 1 in the output of the neuron when the object is in that position and blue regions represent 0 in the output of the neuron when the object is in that position.

and so on, until having the neural actitvity for all the possible object positions around the agent. Thus neural activations are then plotted in the object position which generates them (figure 4.4). Red regions represent high activation and blue regions represent no activation.

With this map we can observe the activation of each neuron for all the situations that the agent can find the object (within a limited range). For instance, when the object is to the left of the agent (see the upper region of the neurons in figure 4.4), the left sensor neuron (neuron 0) is very active (red) and, therefore, the left motor neuron (neuron 1) is inhibited (blue) and the right motor neuron (neuron 5) is excited (red), so the agent turns to the left. Following that movement, the object is in front of the agent and the activity of the neurons is uniform (middle right region of the neurons) and the agent approaches the object. Due to symmetry, the analogous situation happens when the object is located to the right of the agent.

Directional light sensors

As previously mentioned, visual systems in animals evolved from primitive light detectors into directional and spatial light sensors. Mimicking this evolutionary development, we restricted the light sensors to be directional in the simulated mobile agents, and evolved controllers in the same conditions as the last experiment. After restricting the sensory activity to a particular angle, and finding successful agents in 200 generations (compared to thousands when using panoramic sensors), the best evolved controllers were again tested systematically. Typical results of such evolved controllers are shown in figure 4.5.

This time, the agent can sense the object only when the latter is in front of the former, therefore, the agent has to rotate to be able to locate the object. A typical example is shown in figure 4.5. Thus, the agent has to be active, in contrast to the panoramic sensors case where the agent did not need to move to start sensing the object. This is shown by the neural dynamics which oscillate much more than in the previous case (compare

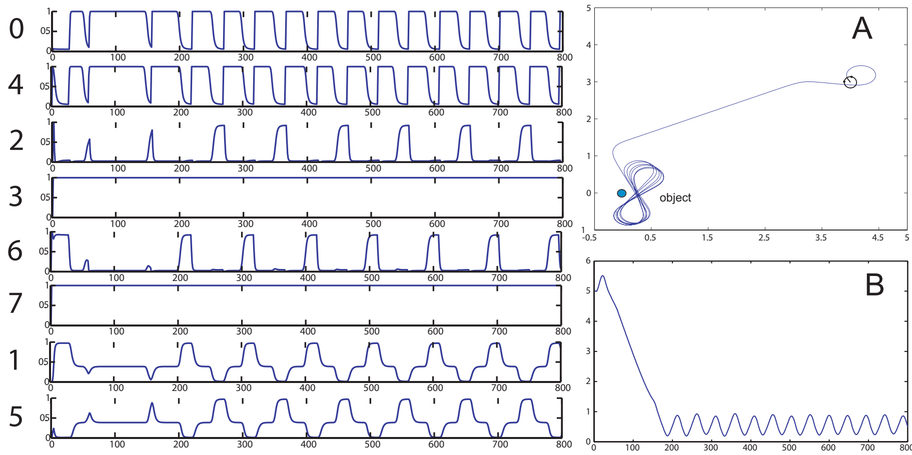

Figure 4.5: Neural activity of an evolved controller of 8 neurons using directional sensors during a test trial. Neurons 0 and 4 are sensor neurons, neurons 1 and 5 are motor neurons and neurons 2, 3, 6 and 7 are interneurons. (A) Positions of an evolved agent during a test run. The object (target) is placed in the centre of the arena (0,0). (B) Distance between the agent and the object during the test run.

sensor neurons 0 and 4 in figures 4.5 and 4.3). However, when using directional sensors, interneurons 3 and 7 are saturated during the test trial, suggesting that these neurons might be redundant (see the figure 4.5).

A simpler controller with six neurons was therefore evolved to perform phototaxis using directional sensors. Successful evolved controllers were found around the hundredth generation. These agents had a similar behaviour to the 8 neuron agents with directional sensors. A typical controller dynamics and behaviour example for 6 neuron controller agents with directional sensors are shown in figure 4.6.

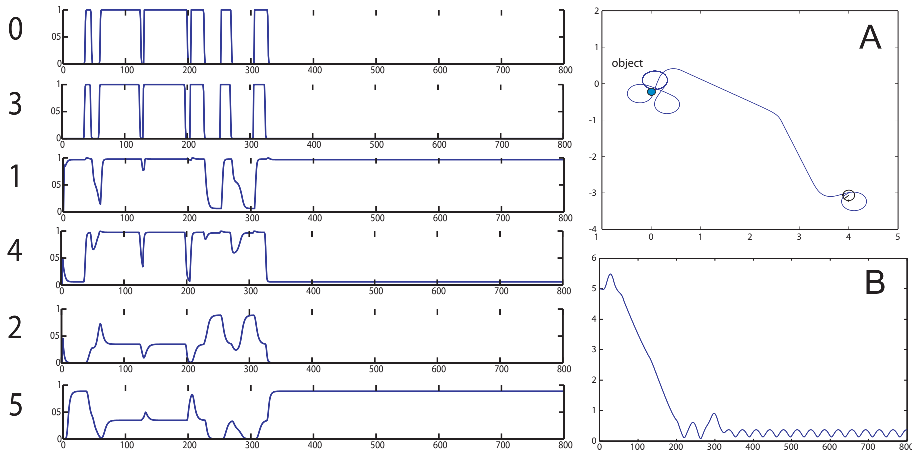

Figure 4.6: Neural activity of an evolved controller using 6 neurons and directional sensors during a test. Neurons 0 and 3 are sensor neurons, neurons 1 and 4 are interneuron and neurons 2 and 5 are motor neurons . (A) Positions of an evolved agent during the same test. The object (target) is placed in the center of the arena (0,0). (B) Distance between the agent and the object during the test run.

With a simple neural controller, it is possible to analyse its neural dynamics and fully explain the phototactic behaviour of the evolved agents. The interaction between the agent and the environment can be described by two general situations: (1) when the object is not within the visual field and (2) when the object is within the visual field. The most important aspect to explain is how the agent "decides" when to turn and navigate to find the object. This is carried out by a modulatory process of sensors and motors done by the interneurons since there are no feedback connections in the controller.

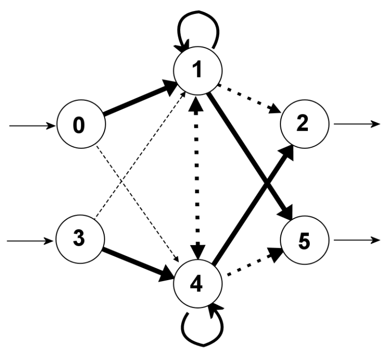

Figure 4.7: Simplified neural network: 6 nodes. Only two interneurons. The dotted line arrows represent inhibitory connections and solid arrows represent excitatory connections. The width of the arrows is proportional to the strength of the connection.

If the object is not within the visual field (situation 1), neurons 1 and 4 play the modulatory role. If the sensor activity is different (due to random initialisation), the interneuron that is more active overcomes the other one due to the inhibitory connections between interneurons (see figure 4.7). And since the interneurons (1 and 4) are connected to the motors, the dominance of one of the interneurons is then reflected to the motors and finally translates into a steady spinning of the agent which brings the object to the visual field (situation 2).

Once the agent is spinning, the sensor which corresponds to the side where the object appears in the visual field is then activated first. This difference in the activation of the sensor neurons will balance the interneurons until an equilibrium in their activity is reached and the agent goes towards the object. For example, if neuron 1 was more active than neuron 4, the left motor would be more active than the right motor, so the spin would be clockwise until the object was within the field of view. At this point the left sensor is increasingly more until an equilibrium is reached between neurons 1 and 4 due to their mutual inhibitory connections (see figure 4.7). This regulatory process also happens to the motors so that the agent starts to go in straight line, which roughly corresponds to the direction of the object. Once the agent has passed the object and the latter is no more within the visual field, the agent is back to situation 1 in which the difference of the sensor activity makes it start the spinning behaviour.

The performance of the evolved controllers for phototaxis is shown in figure 4.8. The averaged population fitness over the first 500 generations for the controllers with 6 neurons (6ND) and directional sensors is higher than both, agents with controllers with 8 neurons using panoramic sensors (8NP) and agents with 8 neuron controllers using directional sensors (8ND). This is because, 6 neuron controller agents with directional sensors are more reactive and stay closer to the object, so they score higher than 8 neuron controller agents, which in contrast, have to unsaturate the sensor neurons and go away from the

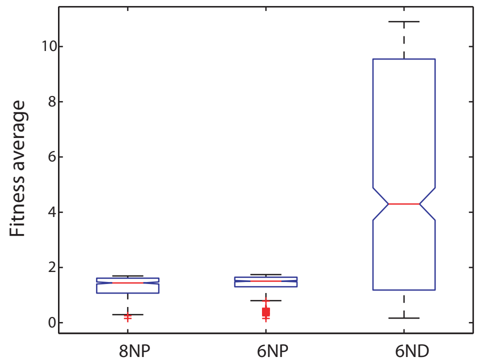

Figure 4.8: Average fitness of the population for the first 500 generations for different controllers and types of sensors. Controllers with 8 neurons and panoramic sensors (8NP), controllers with 8 neurons and directional sensors (8ND) and controllers with 6 neurons and directional sensors (6ND).

object for most of the trial. However, by having panoramic sensors, in general, it is easier to find the object since it can be always sensed, in contrast, 6 neuron agents with directional sensors require to spend time finding the object first (search behaviour) and then approaching it.

Since we tried to find 6 neuron controllers using panoramic sensors but the evolutionary process was not able to find successful controllers in several thousands of generations, therefore it is easier for the evolutionary process to find successful 6 controllers with directional sensors. Additionally, with the directional restriction of the sensors, the reduction of the complexity in the neuro-controller (from 8 neurons to 6) was also beneficial because it was possible to analyse and understand the neural dynamics of the controller and the behaviour of the agent.

The type of sensor used determined a different type of problem to solve to perform phototaxis. For directional sensors, the agents had to find the object first before approaching it. For panoramic sensors, it was required to disambiguate between the two possible positions of the object. That is, the activation of the sensors would be exactly the same for situations where the object is in front or behind but with the same distance from the agent as shown in figure 4.9.

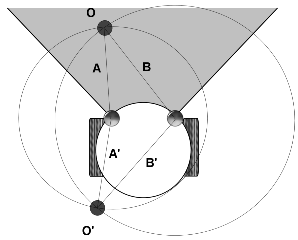

Figure 4.9: Ambiguous situation: the activation in the sensors when the object O is in front of the agent, is equivalent to the activation generated from the object O'. The distance from O to the sensors is the same as the distance from O'. That is A = A' and B = B'.

This experiment shows that an ER approach is viable to find successful controllers for solving a simple visual guided task and it produces an active strategy that uses a simple controller. However, the visual system of the agents is very simple, so we now want to see if the results using such approach can be "scaled" to a more complex visual system such that object recognition is possible.

4.3 Experiment 2: Increasing the complexity of the visual system.

In this experiment I use a simulated camera to acquire the visual information in autonomous agents to perform object recognition. In the previous experiment the ER approach was used to find successful controllers. However, when using cameras, the visual information processing imposes a restriction in time over the evolutionary processes. To overcome this problem, an alternative approach is proposed in this experiment which consisted of using the ER approach and a simulation of the rich visual information to find successful controllers for "object approaching" and "object colour discrimination". The former task requires the agent to approach an object placed in the arena and remain as close as possible to it at the end of a fixed period of time. The latter task requires the agent to discriminate between two different objects by approaching only one of them and remain as close as possible to it at the end of a period of time.

4.3.1 Methods

In order to have a visually guided autonomous mobile agent performing object recognition, it is needed to have a visual system rich enough such that the incoming visual information contains useful visual information, and also, a controller capable of provide the required movements to perform object approaching and object discrimination. However, because we employ an ER approach and the information from the simulated video camera is given in real time, two types of agents were simulated, one with a visual system using a simulated camera, called the "rich simulated agent" (RSA) and another using a simplified visual system called the "simple simulated agent" (SSA). The idea is to find successful controllers using ER for the SSAs, and analyse whether these controllers can be transferred to the RSAs.

Rich simulated agent (RSA)

The RSA has a circular body with radius of 0.5 units and two wheels driven by two independent motors, and a camera on top of its body. The visual system of the RSA has a visual field which is a grey-scale (0-255) region from the simulated camera. This region is 512×32 pixels (see figure 4.10). The visual system has a blob detection mechanism (is described in 3.2.3) and two types of sensors. The blob detection mechanism selects visual subregions of consistent pixel intensity with area in the range 10-50 pixels. Only one 'blob' is selected at any time. In cases where there is more than one blob in the visual field, the visual system selects the blob with the largest area.

Two types of sensor respond to a selected blob. The first is a "location sensor" which is activated by the inverse of the distance (L or R) between the object and the corresponding edge of the visual field (see figure 4.10). The RSA has a left location sensor and a right location sensor. The second sensor type, used only in the object discrimination task, are

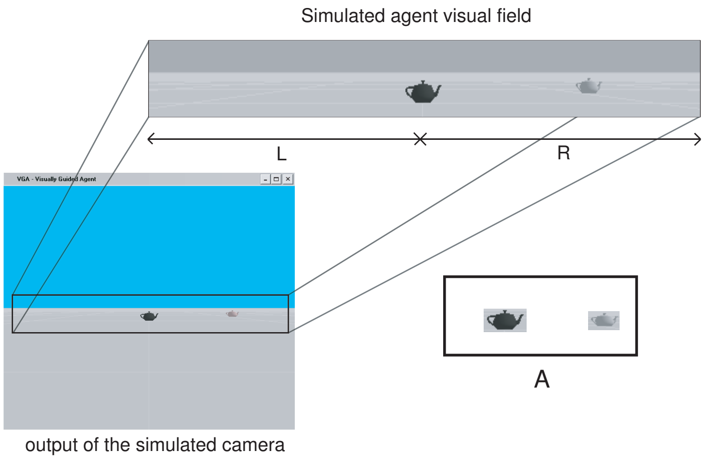

Figure 4.10: Simulated visual system. The visual field of the agent is a region of 512×32 pixels. L is the distance from the object to the left edge of the visual field and R is the distance from the object to the right edge. The inset A in the figure shows the detected blobs (from a distance of 2.5 to the dark object and 3.0 units to the light object) containing the light and dark objects respectively. In this example, the dark object is the largest and so the sensor neurons will respond to this object.

"colour sensors" that return the pixel intensity of the centroid of the selected blob. Due to the fact that the rich visual simulation incorporates directional illumination and reflectance properties of the objects in the arena, the pixel intensity at any time is a complex function of intrinsic properties of the object detected and the reflectance of the object in the corresponding region of the visual field. Although the two colour sensors receive identical input (unlike the location sensors), they may still produce different outputs depending on intrinsic neuron properties (see below).

At the beginning of each evaluation, an RSA was randomly positioned within a region of 12x12 units in an unlimited arena. For object approaching experiments, a visual object (a dark-coloured kettle) was placed in a fixed position in the arena. For the object discrimination task, a light coloured kettle (target) and a dark coloured kettle (distracter) (see inset A in figure 4.10) were placed in the arena in positions (0, -4) and (0, 4), respectively. During the test phase, each trial lasted for 200 time-steps; during analysis of evolved controllers, each evaluation lasted for 800 time-steps.

Simple simulated agent (SSA)

The SSA has a circular body with radius of 0.5 units and two wheels on both sides of the agent, driven by two independent motors. The simplified simulated visual system of this agent has a visual field that is restricted to a region of fixed width V. This region is also limited by two lines originating from the center of the agent extending ± 45 deg from the orientation of the agent (see figure 4.11). It is important to emphasize that this region is spatial, in the sense that it is defined in terms of a subregion of the arena, rather than, as is the case for the RSA, as a subregion of a visual image. This difference means that sensory signals for the two agents will have different dynamical structures. For example, it is possible that a visual object will move in and out of view for the SSA (because of the fixed width V of the visual field) while remaining constantly within view for the RSA.

(One situation in which this may occur is as an agent spins.)

It is also important to notice that there is a spatial region near to the agent where the SSA is blind (the light gray region in figure 4.11). As we describe below, this blind region is important in the explanation of the evolved behaviour of the SSA.

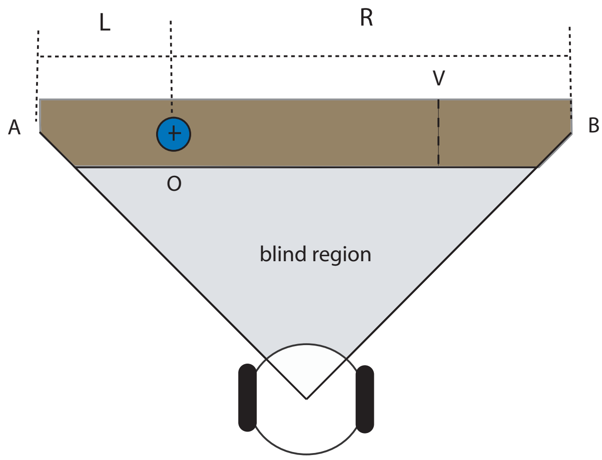

Figure 4.11: Visual field of the SSA: the object (O) can only be sensed if it is within the dark brown region. This region is limited by two lines extending from the center of the agent at ±45 deg from the agent's orientation and a width of V. L and R are the distances between the object and the left and right edges of the visual field, respectively.

As with the RSA, the SSA has two types of sensors which take input from the visual system. The location sensors of the SSA are activated by the inverse of the distance (L or R) between the object (if it is within the visual field) and the corresponding edge of the visual field. The colour sensors of the SSA return a similar value to the pixel intensity of the objects (40 and 130 for the dark and light kettles, respectively) used for the RSA. To deal with the variation in values of colour sensors for RSAs (resulting from changes in reflectance and in intrinsic properties of the selected blob), colour sensors for SSAs were modulated by a random value [-30, 30] (distributed uniformly).

The activation of the sensors in the RSA and the SSA worlds are different. For example, the blind regions in the SSA (in any case where the object is not in the brown region in figure 4.11) does not exist in the RSA (as previously explained). That is, there are instances where the object is not seen by the SSA, but it visible for the RSA for the same position in the arena. Additionally, the colour sensors activity is different, the variation in the RSA sensor activity is given by the reflectance properties of the object and the variation in the centroids detected by the visual system. In contrast, variation in the colour sensor activity in the SSA is given by uniform noise around arbitrary values.

Controller

The controllers for both types of agents were again CTRNNs. As previously mentioned, in a CTRNN, the state y of each neuron i changes in time according to the differential equation:

$$\tau_i \frac{dy_i}{dt} = -y_i \sum_j w_{ji} \phi(y_j + \beta_j) + g_i \cdot I_i$$

where ϕ is the sigmoid activation function, τ is a time constant, β is a bias, and w_{ij} represent connection weights from neuron i to neuron j. The state of each neuron is therefore the integration of the weighted sum of all incoming connections (plus a gain modulated input $g_i \cdot I_i$ for input neurons).

For the object approaching task, the CTRNN consisted of eight neurons, specifically, two sensor neurons, four fully connected interneurons and two motor neurons. For the discrimination task, two more sensor neurons corresponding to the colour sensors were added (see figure 4.12). Parameter values for all neurons were initialised in the following ranges: $\tau \in [0.2, 2.0]$, $\beta \in [-10, 10]$, and connection weights $w_{ij} \in [-5, 5]$. In the object discrimination task, neurons 8 and 9 used $\tau \in [0.2, 10.2]$ and bias $\beta \in [-30, 30]$. All parameter values were shaped by the GA (see below).

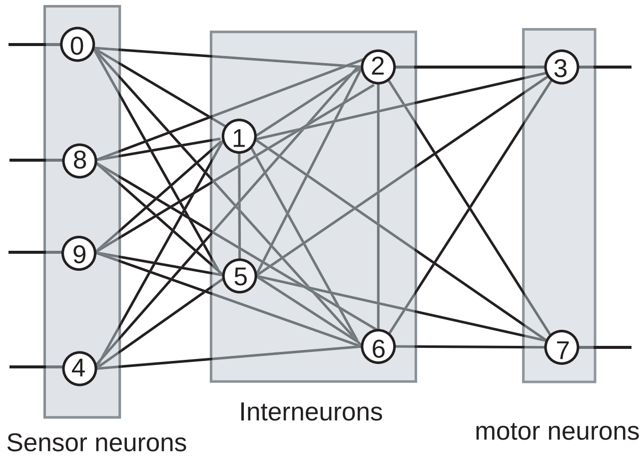

Figure 4.12: Controller. Neurons: 0 and 4 are location sensors; 8 and 9 are colour sensors. Neurons 1,2, 5 and 6 are fully connected. Neuron 3 is the left motor neuron and neuron 7 is the right motor neuron. Note that the colour sensor neurons 8 and 9 were *not* used for the object approaching task.

The controllers were symmetrical (i.e., same parameters were used for each pair of sensor neurons, 0 and 4; 1 and 5; 2 and 6 and so on. See figure 4.12.), except for neurons 8 and 9 which had independent parameters.

Genetic algorithm

The same distributed GA used in the previous experiment was used to evolve CTRNNs to perform the visually guided tasks in this experiment. In this case, the genome of each individual was coded as a real vector of 32 elements for the object approaching controller and 39 elements for the object discrimination controller. For the 32 element vector, 4 elements were used to code the time constants of each neuron, 4 for the bias of each neuron, 2 for the sensor gains and 22 for the weights. Each element was coded as a real number in $[0, 1]$ and linearly scaled according to the parameters previously described in section 4.3.1. For the 10 neuron controller 7 elements were added, 2 for the bias of the two extra sensor neurons, 2 for the time constant, 1 for a sensor gain for these neurons and 2 for the weights. A population of 400 individuals was evolved with mutation probability of 80% for each genotype and 20% for mutation change for each vector element and an elitism probability of 80%.

Two fitness functions $F_1 = 1/d_f$ and $F_2 = 1/d_l - 1/d_d$ were used. F_1 was used for the object approaching task and F_2 for the object discrimination task. In F_1, d_f is the distance from the agent to the object at the end of the trial and in F_2, d_l and d_d are the final distances between the agent and the light and dark objects respectively. The fitness of each individual was calculated as the average across 5 independent trials (of 200 time-steps each).

4.3.2 Results

After several thousands of generations controllers were successfully evolved for both tasks. As mentioned previously, controllers were evolved using the SSA and then tested in both types of agents, SSA and RSA.

Object approaching task

For this simple task successful controllers were found quickly (before 2000 generations). As we can see in figure 4.13B, the agents used an exploratory strategy, first spinning until the object was within the field of view and then approaching the object and rotating around or very close to it.

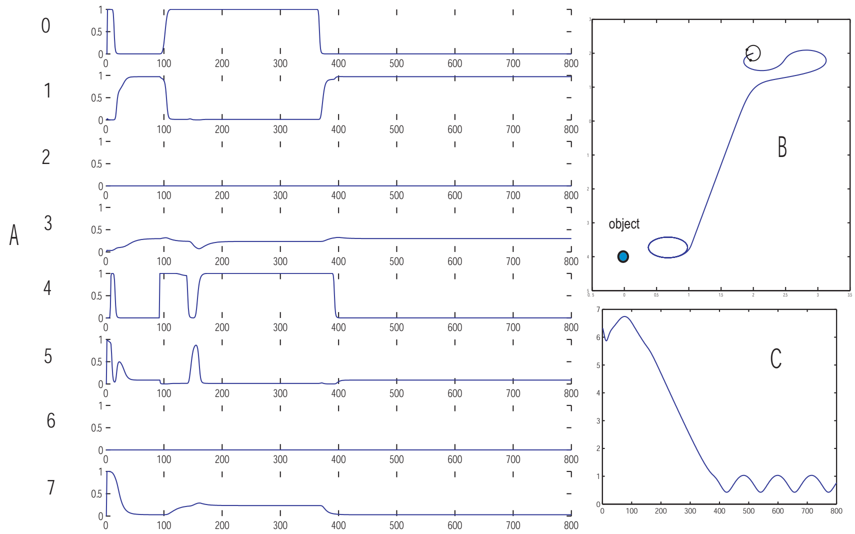

Figure 4.13: Object approaching by an SSA. [A] shows the neural activity during a test trial of 800 time-steps. [B] shows the distance between the agent and the object during the trial and [C] shows the distance between the agent and the object during the trial.

Successful controllers for SSAs were successfully transferred to agents using the rich visual system (RSAs). These evolved controllers also performed the object approaching task successfully (see figure 4.14B). The behaviour of the RSAs was similar to that observed for SSAs: rotate or explore until the object is within the visual field, approach the object and then rotate close to it. In the particular case shown in the figures, the circle described by the trajectory of the RSA at the end of the trial is bigger than that described by the trajectory of the SSA. This observation is highlighted by figures 4.13C and 4.14C, where the distance to the object is shown during the trial.

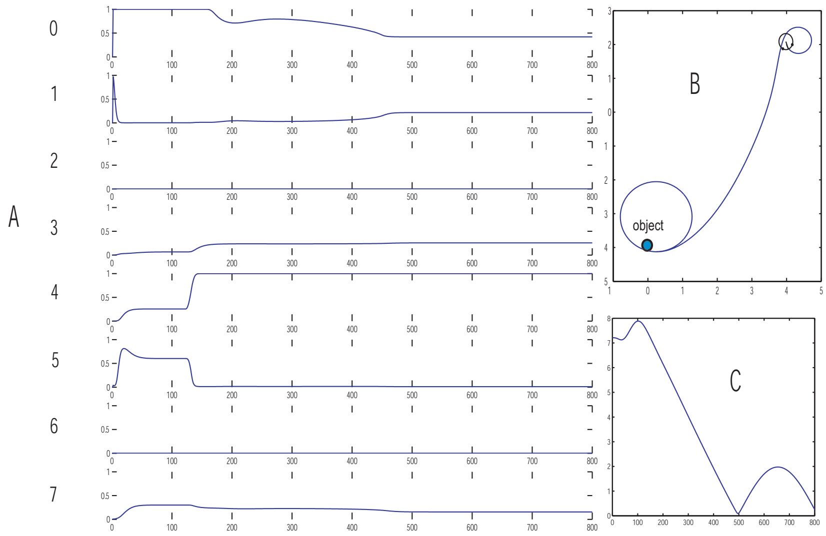

Figure 4.14: Object approaching performed by an RSA using the evolved controller shown in figure 4.13. [A] shows the neural activity during the test trial of 800 time-steps. [B] shows the trajectory of the agent during the trial and [C] shows the distance between the agent and the object during the trial.

Object discrimination task

In this case the task was to discriminate the objects using pixel intensity information. Successful discrimination was reflected by approaching the target object (the light-coloured object). SSAs were successfully evolved to perform this task in 200 generations approximately. Figure 4.15 shows a SSA performing the object discrimination task during a test trial.

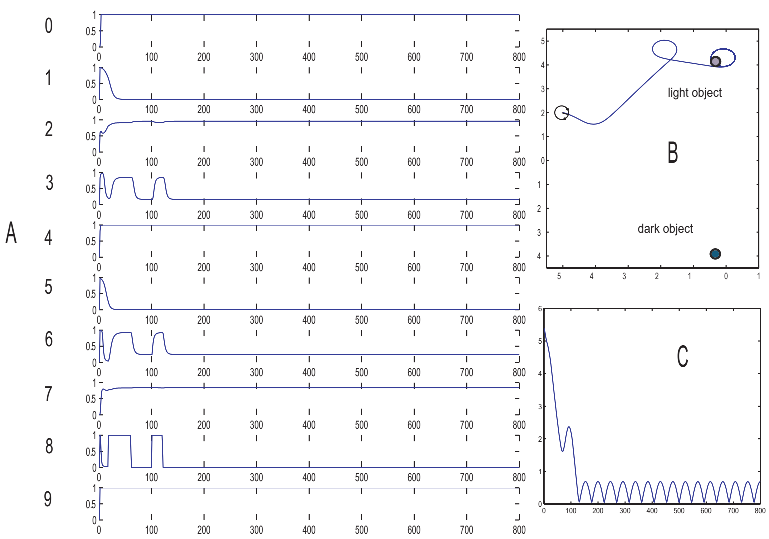

Figure 4.15: Object discrimination performed by an SSA. [A] shows the neural activity during a test trial of 800 time-steps. [B] shows the trajectory of the agent during the trial and [C] shows the distance between the agent and the object during the test trial.

As shown in figure 4.15, the dark (distractor) object is initially within the field of view but the agent nevertheless turns towards the target object and then approaches it. At the end of the trial the agent rotates in close proximity to the target object. The same controller transferred successfully to the RSA. Figure 4.16 shows an RSA performing object discrimination task using this evolved controller. As in the first task, the behaviour of the RSA is similar to that of the SSA. The agent rotates until the object is within its visual field and then approaches it. In the trial shown in figure 4.16, the dark object is closer to the agent at the beginning of the trial but, after a short time, the agent moves away from the dark object and subsequently approaches the target. Note that for the object discrimination task, both SSAs and RSAs stay very close to the target object (compare figures 4.15C and 4.16C).

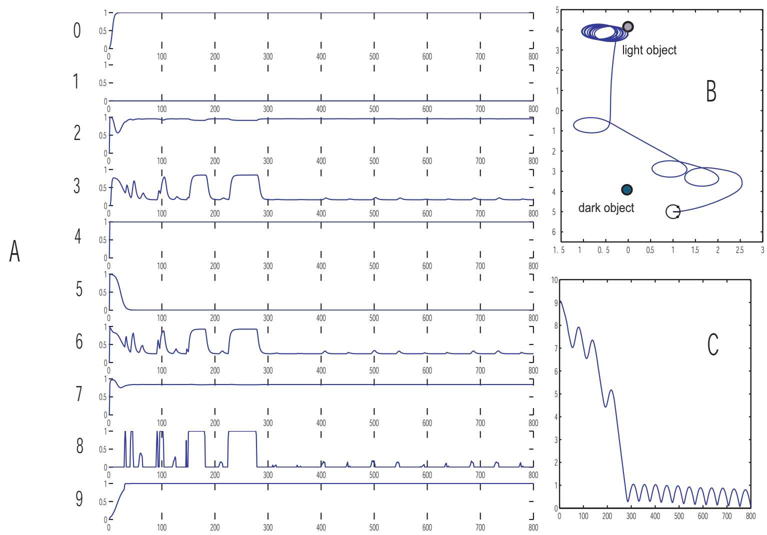

Figure 4.16: Object discrimination performed by an RSA using the evolved controller shown in figure 4.15. [A] shows the neural activity during a test trial of 800 timesteps. [B] shows the trajectory of the agent during the trial and [C] shows the distance between the agent and the light object during the trial.

It is important to emphasize that, for this task, certain aspects of simulation of the colour sensors were critical for the successful transfer of controllers. Specifically, evolutionary runs in which random variance in these sensor values was not incorporated (see section 4.3.1) showed considerably decreased performance when transfer to an RSA was attempted. During attempted transfer in these cases, variance in the RSA colour sensor values (due to the richness of the visual simulation) resulted in these agents approaching both object types equally often.

4.3.3 Analysis

In general the strategies of both SSAs and RSAs can be described as follows. First, agents rotated until an object was within the field of view, then agents approached the object, and finally, agents rotated either close to or around the object, until the end of the trial.

In order to better understand the dynamics of evolved behaviours and the factors underlying successful transfer between simulations, I now examine evolved behaviours in terms of neural activity. For both agent types, the initial rotating behaviour can be attributed to the random initialisation of the CTRNN. This was shown by initialising the neurons uniformly, in which case both SSAs and RSAs navigated in a straight line at an arbitrary heading (data not shown). The approach behaviour of both agent types can be attributed to sensor activation corresponding to an object perturbing the equilibrium point in neural dynamics corresponding to the spinning behaviour. This was shown by testing SSAs and RSAs without any object in the arena (data not shown).

For the object approaching task, neurons 2 and 6 were always constantly saturated for both agent types and therefore can be discarded from the analysis (see figure 4.13A and 4.14A and figure 4.12), leaving only neurons 1 and 5 as modulators of motor neuron activity (see figure 4.12). For the object discrimination task, all the sensor neurons are constantly saturated except for neuron 8 (again for both agent types). Since this type of neuron has a different weight for each connection, it is still able to modulate neuron 6 which in turn is responsible for regulating the motor neurons (see figure 4.15 and 4.16).

The final segment of successful agent behaviour involved rotating close to an object. This behavior was related to the initial rotating (described previously). Once the agents were sufficiently close to the object so that the object was within the "blind region" (see section 4.3.1), they reverted to spinning. In the object approaching task, when this happens, the agent could no longer sense any object and the situation was equivalent to the one where no object was present. For the object discrimination task, once the agent was spinning very close to the target object but was not able to sense it, the agent could still sense the dark (distracter) object (see neuron 8 in figure 4.16A, the small peaks correspond to the dark object and high peaks to the light object) but the activation of the sensor neuron was not high enough to trigger approaching behaviour. This situation is not shown in figure 4.15 because the agent is spinning too far away from the dark object to be able to detect it, however the same situation applies to both SSAs and RSAs.

In general, the behaviour of the evolved controllers shows that despite the differences in the dynamical structure of sensory signals between SSAs and RSAs, evolved controllers transferred successfully from one to the other. As the neural analysis shows, this transfer was possible because evolved agents relied on consistent features of sensory activity, and not on those aspects that varied between the agent types (see section 4.3.1).

4.4 Conclusion

In the first experiment of this chapter, controllers were evolved to perform phototaxis using panoramic or directional sensors. It was shown that by restricting the sensory system, a simplification of the neural controller, can be achieved. In particular, it was possible to evolve a controller with less neurons to perform the same task. By this reduction of the dimensionality of the problem, it was easier to analyse the neural dynamics of the controller and the behaviour of the evolved agent. The behavioural result of this restriction in the visual sensors improved the performance of the agents for this task by producing more reactive agents. Even when the visual system used by the agents is very simple, the results of this chapter stress the relation between sensors, controllers and motors from an active perception perspective.

The directional sensors employed in this experiment correspond to a simple form of attentional techniques in primitive visual systems. This experiment is important because it demonstrates how an active approach can reduce the complexity of an autonomous agent. These results encourage the exploration of more complex visual systems using this methodology in order to study the role of active vision approaches to object recognition in mobile agents.

The second experiment of this chapter consisted of an extension of the approach used in the previous experiment to study autonomous mobile agents using more complex visual systems. It was shown that evolved controllers for agents using a simplified visual system (SSAs) could be successfully transferred to agents using more complex visual information (RSAs). The behaviour of both agents (SSAs and RSAs) for object approaching and discrimination was fully explained by analysing the dynamics of their neural activity. In this way, it was shown that the complexity gap between SSAs and RSAs was crossed.

This work demonstrates how the ER approach to study autonomous mobile agents with a rich visual system could be used. The development of increasingly complex simulations blurs the distinction between simulation and reality. A hierarchy can be envisaged in which controllers are initially evolved in simple simulations and then are incrementally refined in progressively more complex simulations until final deployment in a real world environment. In addition, rich simulations offer the possibility of exploring detailed agent-environment interactions which do not exist in real-world situations, thereby supplying potentially valuable comparison conditions for understanding mechanisms of adaptive behaviour. Future work in this area could usefully consider the development of *minimal* simulations of rich simulations, in the sense described by Jakobi (1998). Minimal simulations incorporate extremely high levels of noise in specific loci in order to ensure that evolved controllers cannot rely on these aspects of agent-environment interaction. This method might extend the 'complexity gap' between simulations that can be feasibly traversed by evolutionary approaches.

Chapter 5

Active acquisition of visual information

5.1 Introduction

In chapter 3 we show that the complexity of the HMAX model can be reduced through the exploitation of an active mechanism. We then show that, in principle, we can generate controllers through ER which can produce simple movement patterns. The next stage, and the goal of this chapter is to determine whether the HMAX and RBF models can exploit the variation in the visual information through simple movements and, specifically, how the variation in the training and testing views is exploited by the models. In particular, we want to investigate whether the variation in scale and/or rotation provided through simple movement strategies impacts the models differently. Following this, we study the conditions under which the models are robust to noise in the visual system.

In this chapter I therefore employ a simulated agent to acquire the training and test views through simple movements which will be analysed by the the RBF and HMAX models. I first analyse the potential of this approach by examining the training views acquired by the agents. In particular, I assess robustness to noise when more training views are used, since it has been reported that increasing the number of views increases robustness (Edelman, 1997). I then investigate the performance of the models when a different trajectory is used during testing than the trajectory used to acquire the training views. Finally, I assess whether or not the models can exploit movement to improve the recognition performance by examining different test trajectories.

5.2 Methods

This experiment consisted of training and testing the models under different conditions. During the training phase, the models were trained using views collected while the agent was traversing a circular trajectory (figure 5.3). The conditions during training were investigated by both increasing the number of views used and adding random noise to the agent's visual system. During the testing phase, the models were tested while the agent was following one of two different trajectories (figure 5.6). In this chapter, the views used during the training of the models will be referred to as training views, similarly, the views obtained during the testing trials will be referred to as testing views. Views that were processed by the HMAX model will be called HMAX views and, similarly, if processed by the RBF model, the views will be called RBF views.

5.2.1 Agent, arena and objects

The experimental set up described in this chapter is very similar to the one used in chapter 4. The simulated agent is the same, except that the visual system of the agent is the one described in chapter 3. Specifically, the visual system consists of an analysis module and a classifier module. The former can be either the HMAX model or the RBF model. The latter is a RBFN with centres specified by the training views (see details in sections 3.2.1 and 3.2.2). Additionally, the visual system employed the blob detection mechanism (BDM) described in section 3.2.3 to attend to selected regions in the visual field of the agent. Blobs with larger area were attended to first. The patch extracted with the blob detection mechanism was resized to a uniformly sized (60×80 pixel) grey image (see figure 5.4 for an example of resized blobs detected).

The arena is an unlimited planar surface containing two simulated objects, a teapot and a bolt-like object (see figure 5.4) which the models are trained to discriminate. Different views of these objects (referred to as views or training views, depending on whether or not they were processed by the models) were used to train the models.

Analysis module. The analysis module processes visual information coming from the BDM (explained in section 3.2.3). These views are processed by either the HMAX or the RBF model (previously described in section 3.2.1). The RBF model emulates simple cells in the primary visual cortex, V1, based on the function of receptive fields implemented by using Derivative of Gaussian filters with different orientations and sizes. The RBF model uses four different sizes of square filters with sides of 7, 11, 15 and 21 units and 0, 45, 90, and 135 degrees of orientation. There are therefore 16 different filters in total with outputs responding to oriented 'edges' at different spatial scales. Therefore, this model responds only to a collection of simple primary features. In contrast, the HMAX model proposed in (Riesenhuber and Poggio, 1999b) is a hierarchical model resembling the ventral pathway in the visual cortex. Briefly summarised (and already mentioned before in this thesis), the HMAX model consists of four layers (S1, C1, S2 and C2) resembling simple and complex cells in the ventral pathway. Units in S1 would correspond to simple features detected by the different filters of the RBF model. The next layer C1, responds to the most salient features in S1 at each orientation and spatial scale. It achieves this by applying max pooling operations (extracting the most salient features across the different orientations and spatial scales) over the selected features in S1. The next layer, S2 combines the output of C1 into a higher order features sets which are passed into C2 where the outputs are again max pooled to produce a vector of the dominant features detected along the hierarchy. By virtue of its hierarchical structure, this model shows a degree of translation and scale invariance.

Classifier module. The classifier module is based on the work of Edelman and Duvdevani-Bar (1997); Poggio and Edelman (1990). The classifier employed here is the same as described previously in section 3.2.2. Briefly, the classifier module uses view tuned units (VTU) to recognise objects. There is one VTU for each object. Each VTU is trained to respond so that it responds strongly to test views that are similar to the training views of the object. Each VTU (see figure 5.1) corresponds to a set of radial basis functions (RBF unit). A RBF unit is a Gaussian function G centered on each view c_i collected during the training phase. The response of each RBF is given by $G(c_i, v) = e^{-\|c_i - v\|^2/\sigma_i^2}$ where v is the vector that is being classified and c_i are the centers, which were located at every training view.

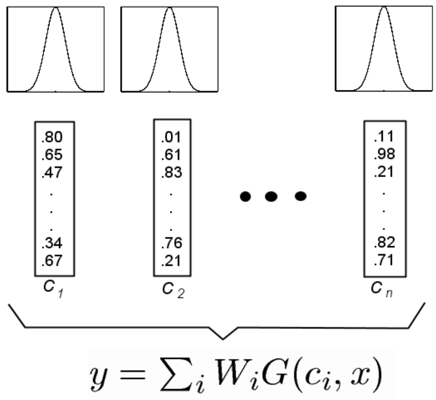

Figure 5.1: View Tuned Unit (VTU): each view vector c_i is the centre of a Gaussian function. The more similar a vector x is to a centre, the stronger the response of the unit.

The response y of each VTU for a test vector x is given by $y = \Sigma_i W_i G(v_i, x)$, that is, y is a linear combination of weights W_i and $G(v_i, x)$. The optimal weights W_i are computed using an inversion matrix procedure (the details were described previously in section 3.2.2 and also in (Orr, 1996)).

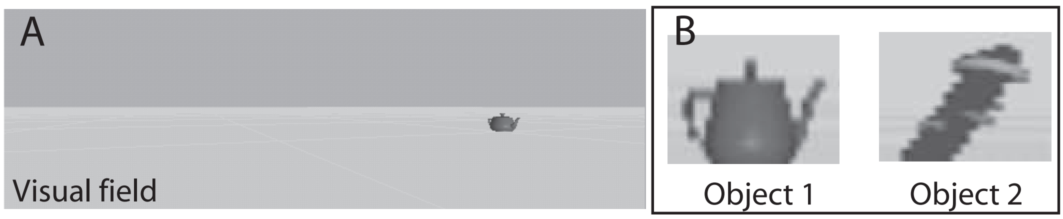

Figure 5.2: (A) Visual field of the agent: shows object 1 in the field of view. (B) Sample views of object 1 and object 2: object 1 is a rounded object so it does not have a significant variability to rotation, in contrast, object 2 has a significantly higher variability to rotation due to its vertical inclination.

5.2.2 Training phase

The models were trained under one of four scenarios, namely using either eight or sixteen training views of each object in the absence or presence of noise in the BDM (see figure 5.4 for an example of the training views for scenario 1). The training views were obtained while the agent was moving around the object following a circular trajectory of 2.5 units of distance to the object, this trajectory is referred to as the training trajectory (see figure

5.3). In order to measure the robustness of the models to noise in the views, a random variation was added to the coordinates of the centroids (x, y) of the blobs detected. The noise was a uniform random variable in the range $\delta \in [-3, 3]$, making the new centroid $(x + \delta_1, y + \delta_2)$. Table 5.1 shows the conditions employed for each of the four scenarios in the training phase. The noise was added to the centroid of the blob detected (after being processed by the BDM) and not by the movement of the agent because the visual variation induced by the motion of the agent was likely to be corrected by the BDM.

scenario	description
1	8 views, no noise
2	8 views, noise
3	16 views, no noise
4	16 views, noise

Table 5.1: Training scenarios for the embodied comparison of the models. The second column in the table shows the number of views that the models used and whether or not noise was added to the centroid of the blob detected by the BDM.

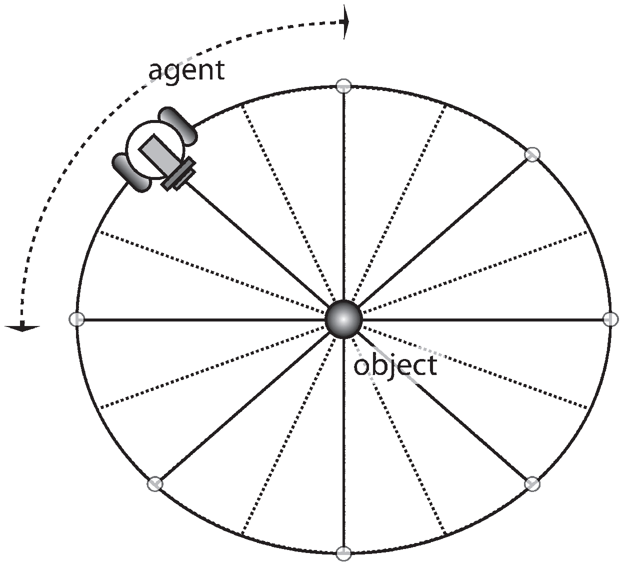

Figure 5.3: Training trajectory. The agent follows a circular trajectory while collecting the training views. The number of views used (8 or 16) for each object determines the positions around the object. Solid lines show 8 different positions where the snapshots (training views) are taken. Similarly, dotted lines show the case where 16 training views are taken.

Figures 5.4 and 5.5 show the training views for scenarios 1 and 3 respectively. Note that the views of object 1 are more similar to each other than in the case of object 2. In scenarios 1 and 2, 8 training views were used. In scenarios 3 and 4, 16 training views were used. The sizes of the images in figures 5.4 and 5.5 were scaled so that the examples could be presented in this thesis.

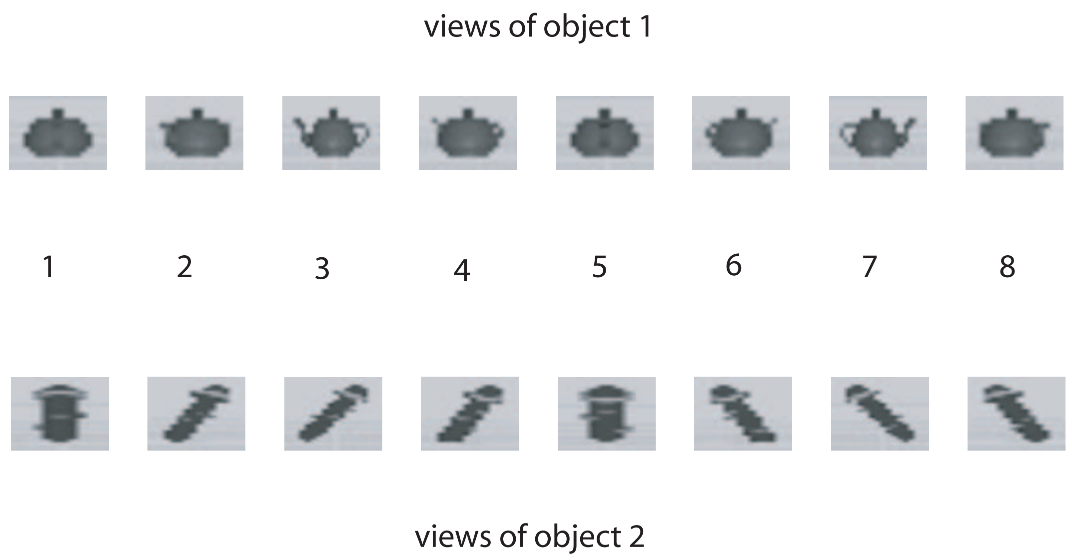

Figure 5.4: Training views for scenario 1. 8 views per object. Object 1 is a teapot and object 2 is a bolt-like object. Every view is 80 ×60 pixels.

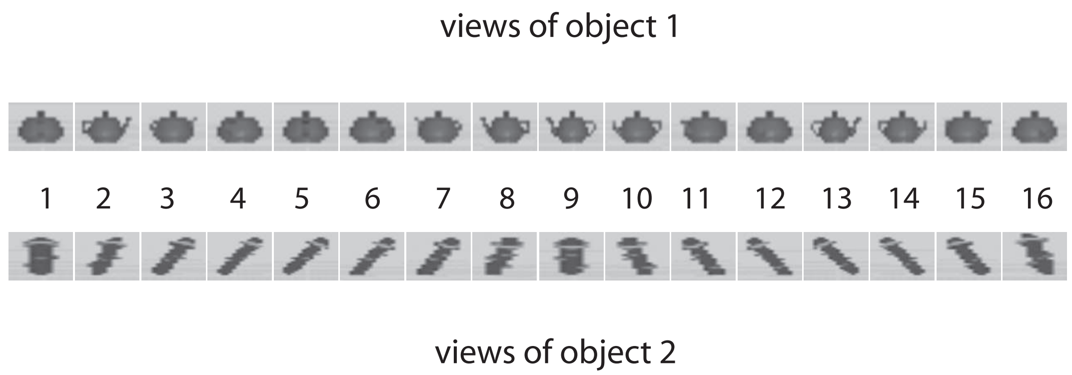

Figure 5.5: Training views for scenario 3. 16 views per object. The sizes of the images are the same as the ones presented in the previous figure.

5.2.3 Testing phase

Both the RBF and HMAX models were tested using one of two trajectories, both different from the training trajectory (see figure 5.6). In trajectory 1, the two objects are present in the arena. In contrast, in trajectory 2 only one object is present at a time. For trajectory 1, the test trial consisted of 200 time steps during which the point of view and the distance to the object changed continuously during the trial. In contrast, trajectory 2 was 140 time steps long and while the distance to the object changed, the point of view was constant. This discrepancy in the way the point of view changed allowed us to test the models in different ways: rotation and scale invariance were both tested during trajectory 1, while in trajectory 2 only scale invariance was tested.

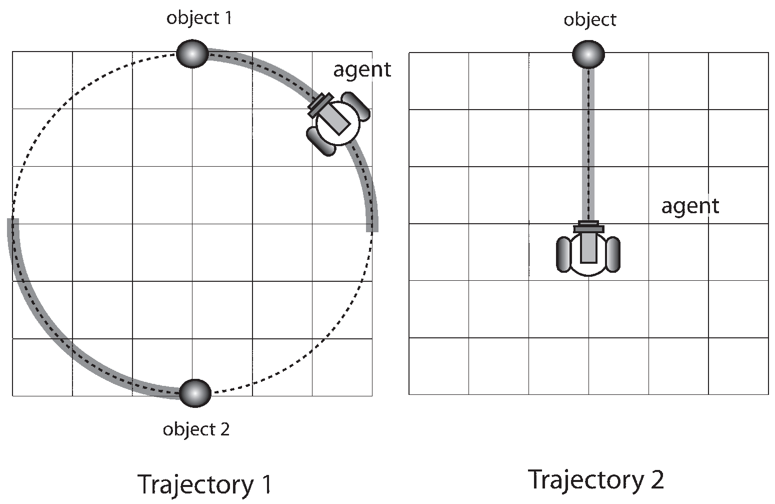

Figure 5.6: Trajectories followed by the agent during the testing phase. In trajectory 1, the agent approaches the two objects following an arc trajectory (dotted line). The objects are within the visual field in the shadowed regions in the trajectory. The initial position of the agent is (3,0) and the positions of the objects 1 and 2 are (0,4) and (0,-4) respectively. In trajectory 2, the agent approaches the objects following a straight line (dotted line). The object is always within the field of view. The initial position of the agent is (0, 0.5) and the position of the object is (0, 4).

A final difference is that while the object is constantly within the field of view in trajectory 2, in trajectory 1, there are two periods where the objects are within the field of view. Period 1 is the first grey segment of the trajectory where object 1 is within the field of view. Period 2 is the second grey segment which represents the time when object 2 is within the field of view (see the grey regions in trajectory 1 in figure 5.6).

The blob detection mechanism used in the training phase was also used to extract patches in the visual field during the testing trials. However, in contrast with the training phase, no noise was added to the centroid of the blobs detected.

5.3 Results

In this section I analyse the way the HMAX and RBF models exploit the differences in the training views acquired through the agent's movement. I investigate how this exploitation affects the performance of the models when they are tested using the two different testing trajectories. First, the performance of the models are presented when tested in trajectory 1 and then the models are tested using strategy 2 (see figure 5.6). For both cases, the models were trained under the conditions described for each of the four different scenarios (see table 5.1). After that, a study of the similarity between the training views of the objects is presented. The study of the similarity of the views is important because the difference between views determines the responses of the classifier module, which is the recognition response of the system. Following that, an analysis of the model activity for each of both cases is presented.

Model performance for trajectory 1. The number of correct guesses made by the RBF and HMAX models when using trajectory 1 for each scenario is presented in table 5.2.

Scenario	RBF	HMAX
1	105	63
2	84	64
3	100	63
4	90	66
mean	94.7 ± 9.5 std	64 ± 1.4 std

Table 5.2: Number of correct guesses by the RBF and HMAX models for each scenario (out of 110 presentations during the test phase) when tested using trajectory 1. The RBF model performs better than the HMAX model in the four scenarios.

With the increase in the number of views used in the training phase, the performance of the RBF confirms that this model shows better robustness to noise in the BDM than when trained with less views (see the difference between the performance for RBF in scenarios 1 and 2 in contrast with the difference of the performance in scenarios 3 and 4). The performance of the HMAX model is lower than the RBF model for most of the cases in the different scenarios for trajectory 1 and the performance of this model is practically unaffected by the number of views or the presence of noise used during training. The difference in the performance of the models when tested using trajectory 1 suggests that the HMAX model suffers when no scale variance is provided during the training phase when the model is tested with significant changes in scale and rotation.

Model performance for trajectory 2. When using trajectory 2 during testing, the performance of the RBF is similar to the previous case. However, the performance of the HMAX model shows an increase compared to the previous case.

Even when, in general, the performance of the RBF model is better than the HMAX model for trajectory 2 (see the average performance of the models for trajectory 2 and the different scenarios in figure 5.7), the difference between the models' performance is significantly smaller than when tested using trajectory 1. Therefore, the performance of the HMAX model is better when the point of view is maintained constant than when the point of view is changing during testing. This result suggests that the HMAX model is exploiting the scale invariance provided by the hierarchical layers and the attentional mechanism when only scale variation (no rotation) is present during the testing phase.

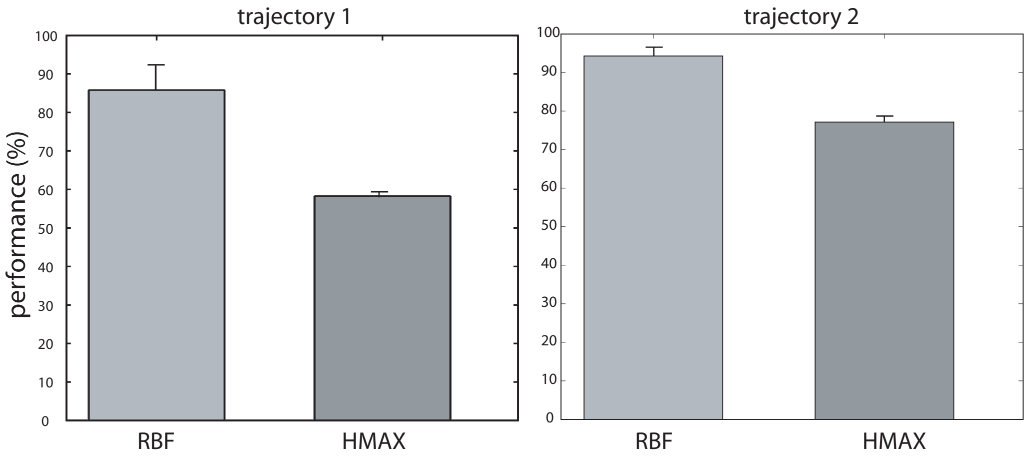

Figure 5.7: Average performance (%) of the RBF and HMAX models over the four scenarios when tested using trajectories 1 and 2. On average, the performance of the RBF is better than the performance of the HMAX model.

5.3.1 Similarity maps

In order to study how the models exploit the differences in the training views under the four scenarios described previously, I present the difference between the views before and after being processed by the models. This difference between the training views of the objects is expressed in what will be referred to as similarity maps.

Measuring the difference (or similarity) between views is important because this determines the response of the classifier module since the response of each VTU is given by a gaussian function centred on the training views of the object (see the details of the classifier module in chapter 3). Therefore, by measuring the difference between the views we can have an idea about how the model would respond when these views are processed.

A similarity map is a diagram representing the similarity between views of the objects (see figure 5.8). The x and y axes represent the views. Every point in the map (i, j) has an intensity defined by the distance between view i and view j. The bluer a point, the smaller the distance is between the views (the diagonal is zero since the distance between the same view is zero). The distance is Euclidean norm of the difference between the views. The first half of the axis in the diagram corresponds to the views of object 1 and the second half to the views of object 2. The map is divided into four main regions, the lower left (region 1) region shows the similarity between the views of object 1, lower right (region 2) and the upper left (region 3) regions show the similarity between views of object 1 and views of object 2, and finally, the upper right (region 4) region shows the similarity between the views of object 2.

Figure 5.9 shows three similarity maps. In the first case (A), the similarity between the views (image views) before they are processed is presented. The second case (B) presents the similarity between the views after being processed by the RBF (RBF views). The last case (C) shows the similarity between the views after being processed by the HMAX model (HMAX views). The blue tones represent high similarity while the red tones represent high disimilarity (large distance or difference). Each subsequent similarity map figure within this section will display the corresponding similarity maps in this order (image

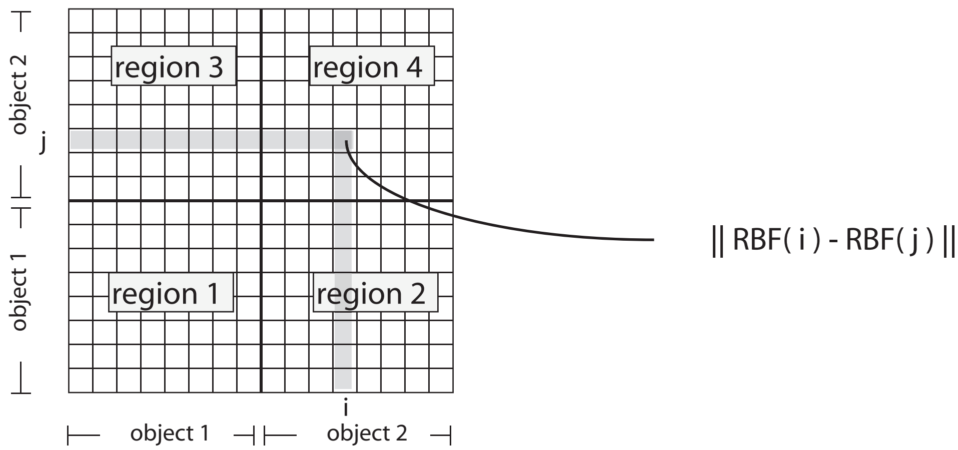

Figure 5.8: Similarity map diagram. Every point in the similarity map (i, j) represents the distance (Euclidean norm) between the views i and j (in this case, after being processed by the RBF model and using 8 views per object). This diagram also shows the regions in a similarity map.

views, RBF views and HMAX views). The separability of the objects (discrimination) is given by the contrast in the maps. As views of the same object should be closer to each other than views of different objects, we would like to see bluish colours in regions 1 and 4 and reddish colours in regions 2 and 3.

Similarities amongst the training views

Figure 5.9A shows the similarity map for the raw training images for scenario 1 (8 views, no noise). The images of object 1 are more similar amongst themselves (region 1), than the images of object 2 (region 4) (the more blue the area, the more similar the views are). This is because object 1 changes less in consecutive training views as the agent rotates around it, than object 2 (since this object has an orientation from the vertical axis). The right lower area corresponds to the similarity of views between the two objects (region 2 is equivalent to region 3). This region, therefore, shows the 'separability' of the objects: the more red the area, the more different the views are, making it easier to discriminate between them (note that due to the symmetry, this area is reflected across the $y = x$ line so this is equivalent for the left upper area of the map).

Figure 5.9B shows the similarity between the views after being processed by the RBF model (RBF views) for scenario 1. In this case, the similarity of the processed views by the RBF model is decreased and smoothed when compared to the similarity in the images 5.9A. In particular, the distance between the RBF views of both objects increased (regions 1 and 4 have more greenish and yellowish tones) and the similarity between the views of object 1 and views of object 2 (regions 2 and 3) are more uniformly reddish than in figure 5.9A, so their separability increased. Figure 5.9C shows that the similarities amongst the views in regions 1 and 4 are generally increased (showing uniform blueish areas) after being processed by the HMAX model (HMAX views). In contrast, for regions 2 and 3, the similarity of the HMAX views is decreased (showing reddish areas). This means that, the separability is increased. Note that the specificity of the maps is different, for example, in the RBF similarity map (B), the point (5,1) for object 1 (region 1), is blue, meaning that

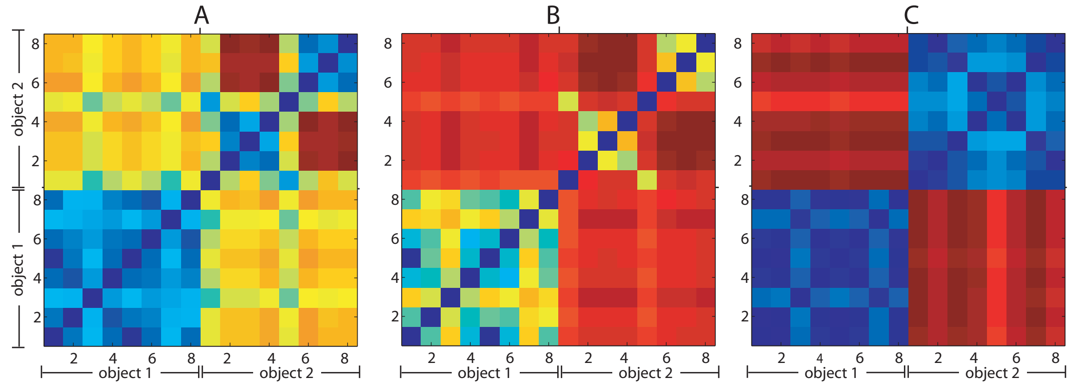

Figure 5.9: Similarity map for scenario 1: (A) shows the similarity map for the images. (B) shows the similarity map of the RBF views (C) shows the similarity map for HMAX views. The blue tones represent high similarity while the red tones represent high disimilarity (difference).

these two views are more similar in comparison with the rest of the views of object 1 (see training views in figure 5.4). The specificity is lost in the HMAX similarity map.

Model	R1	R2	R3	R4
		Scenario 1		
IM	13.58	40.72	40.72	34.69
RBF	28.71	56.38	56.38	47.67
HMAX	6.94	60.80	60.80	10.78

Table 5.3: Sum of the normalised similarities for each region of the maps in figure 5.9. The values in this table were calculated by $\sum_{ij} sim^r_{ij}/max_r(sim)$, where sim_{ij} are the similarity values in the region r and $max_r(sim)$ is the maximum similarity value in region r.

Table 5.3 shows the sum of the normalised similarities for each region in the similarity maps for scenario 1. In this case, the similarity of the regions shows that in the case of HMAX, the similarity of the views was increased in regions 1 and 4 and decreased in regions 2 and 3 which increases the separability of the objects.

For scenario 2, when noise was introduced in the centroid of the blobs detected and 8 views were used, the maps were less uniform areas than in the previous scenario. The similarity of images was increased in general (see blueish and greenish lines in figure 5.10A). After applying the RBF model in scenario 2 (figure 5.10B), the similarity between the RBF views was smoothed and decreased in comparison to the image map (in a similar way to scenario 1). However, the distinction between regions is less evident in this scenario due to the noise. Since the centroid of the objects was randomly shifted, the separability of the objects decreased (see region 2 and 3). However, the similarity between the HMAX views of different objects is increased in scenario 2 (see figure 5.10C). The red lines in the HMAX map correspond to views were the noise in the centroid of the BDM made the object to touch the edges of the image (data not shown).

The sum of the similarities in the four regions for scenario 2 is shown in table 5.4. In this case, the noise affects the separability of the objects.

The results for scenario 3, which used 16 views for each object with no noise (figure 5.11), are similar to scenario 1 . The images of object 1 are similar amongst each other

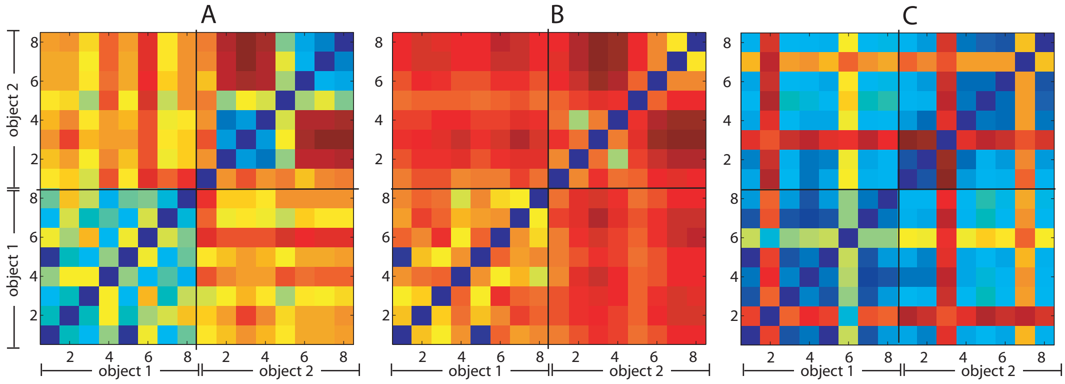

Figure 5.10: Similarity map for scenario 2: (A) shows the similarity map for the images. (B) shows the similarity map of the RBF views. (C) shows the similarity map for HMAX views. For this scenario, the noise in the centroid of the blob detected makes the maps less uniform (compared to scenario 1).

Model	R1	R2	R3	R4
		Scenario 2		
IM	26.52	45.75	45.75	37.49
RBF	39.45	54.37	54.37	47.64
HMAX	22.68	33.74	33.74	28.62

Table 5.4: Sum of the normalised similarities for each region of the maps in figure 5.10.

(blueish areas in region region 1). In contrast, only the first views of object 2 are similar amongst each other, and the last images of object 2 are similar amongst each other (see region 4 in figure 5.12A). This is because views 2-8 of object 2 share the same orientation (see views of object 2 in figure 5.5), and views 10-16 of object 2 share the same orientation. Finally, the similarity between the images of object 1 and object 2 is low (yellow areas in regions 2 and 3) except for image 1 and 9 of object 2 (which are the views when object 2 appears more rounded, and therefore, more similary to object 1). That is because, by increasing the number of views, the similarity between the views increases since the object does not change significantly within consecutive views.

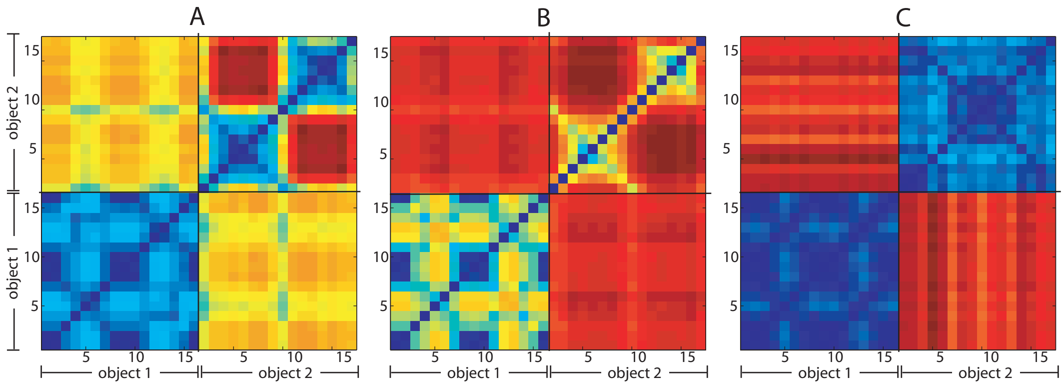

Figure 5.11: Similarity map for scenario 3: (A) shows the similarity map for the images. (B) shows the similarity map of the RBF views (C) shows the similarity map for HMAX views.

The RBF map in this scenario (figure 5.11B) is more uniform than the image map (A).

As before, this is because the RBF mapping makes the difference between the views more uniform because the contrast in the features detected by the RBF changes more uniformly than the images themselves. This map shows that the separability of the objects is also increased (see that the yellowish lines have disappeared). In the HMAX map case (C), again, the separability of the objects is increased. This map shows very distinctive regions (see regions 2 and 3 in figure 5.11C).

Model	R1	R2	R3	R4
		Scenario 3		
IM	55.41	166.13	166.13	147.50
RBF	115.62	226.21	226.21	198.49
HMAX	12.17	58.83	58.83	56.28

Table 5.5: Sum of the normalised similarities for each region of the maps in figure 5.11.

Table 5.5 shows the sum of the normalised similarity values for each region in the maps for scenario 3. In this case, the values show a large difference of the similarities between regions. Specially for the similarity map for HMAX (similarly to scenario 1).

For scenario 4 (see figure 5.12), the models were trained with 16 views and noise was introduced to the position of the centroid of the blobs detected. The maps in this scenario show less clearly separated regions compared to the maps in scenario 3, particularly for regions 2 and 3 and region 4. There is no uniform reddish area in regions 2 and 3 anymore (see the greenish and yellowish lines figure 5.12A).

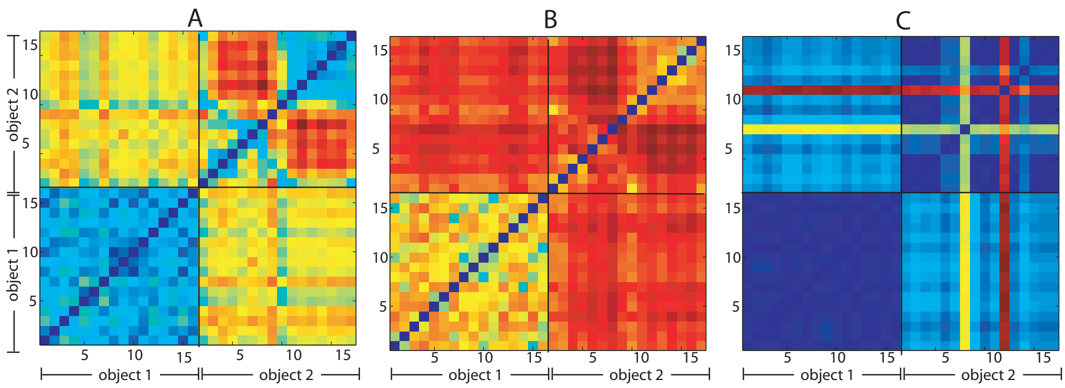

Figure 5.12: Similarity map for scenario 4: (A) shows the similarity map for the images. (B) shows the similarity map of the RBF views (C) shows the similarity map for HMAX views.

Even with the presence of noise and more views, the RBF model makes regions 2 and 3 more uniformly reddish than the images (compare regions 2 and 3 in figures 5.12A and 5.12B). This is not the case for the HMAX views where the reddish regions are turned into blue, only for two views there are reddish lines (views 7 and 11 of object 2) in figure 5.12C). These lines corresponded to views where the noise added to the centroid made the object touch the edges of the image. Again, it seems more likely to have discrimination ambiguities in the HMAX model than in the RBF model for this scenario.

Table 5.6 shows the sum of the normalised similarity values for each region in the maps for scenario 4. In this case, the similarity values reflects the same observation made from

Model	R1	R2	R3	R4
		Scenario 4		
IM	73.66	156.60	156.60	146.75
RBF	155.67	216.87	216.87	203.00
HMAX	20.22	81.66	81.66	63.59

Table 5.6: Sum of the normalised similarities for each region of the maps in figure 5.12.

the similarity maps that shows that by increasing the noise in the images, the separability of the objects is decreased, the sum of the normalised similarity values of the regions is closer amongs each other in comparison with scenario 3.

The overall picture from these results is that when the point of view of the agent during the testing phase is similar to that in training, the performance of the models would be similar to the performance shown in the similarity maps. However, it is important to note that this only would happen if the distance between the agent and the object was kept constant. In the next section, I present a more dynamic and realistic scenario for a mobile agent, where the distance and the point of view change continuously.

5.3.2 Testing the models using movement trajectories

The difference in the similarity of the training views for the RBF and HMAX models suggests that the RBF model should have a better discrimination performance than the HMAX model when the test views are similar to the training views (see figure 5.7). In this section the models trained in the four scenarios are tested while the agent follows one of two different trajectories (see figure 5.6).

Testing using trajectory 1

The response of the models for each time step for trajectory 1 in scenario 1 is shown below in figure 5.13. During the test trials for trajectory 1 (see figure 5.6), object 1 was present in the field of view during the first 55 time steps (period 1). After that period, no object was present in the field of view until time step 125 when object 2 appeared (period 2). Therefore, if the models recognise the objects correctly, during period 1 the signal for object 1 should be stronger than the signal for object 2. In contrast, during period 2 the signal of object 2 should be stronger than the signal for object 1.

Figure 5.13 shows the activity of the models tested using trajectory 1. In this case, as expected, the RBF model recognises both objects correctly for the majority of the time steps. The difference between the signals for object 1 and object 2 for both models for scenario 1 is presented in figure 5.13B. The difference between RBF signals is positive in the first period and negative in the second period. In contrast, for the HMAX model, object 1 is correctly recognised at the beginning of period 1 and object 2 is incorrectly recognised for most of period 2 (the difference in HMAX signals for object 1 and object 2 is negative for period 2 in 5.13B). Note that during the last 10 time steps of each period (approx. from time step 45 to 55 and 170-180), the object appears only partially (or parts of it are detected by the blob detection mechanism) due to the short distance between the agent and the object. This is why the performance drops at the end of both periods for

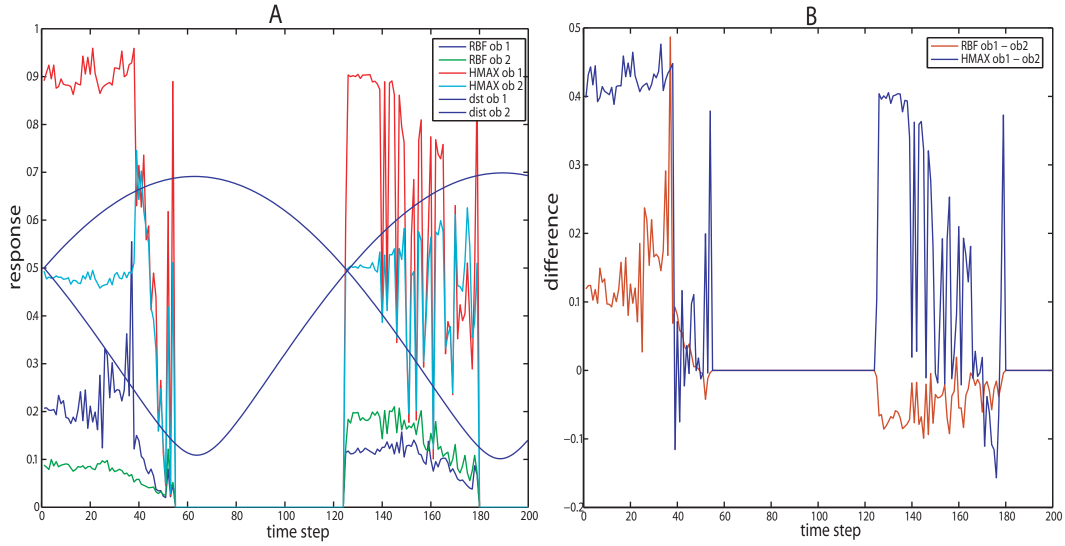

Figure 5.13: ((A) Recognition signals of the two models for trajectory 1 for scenario 1. (B) Difference between the recognition signal of object 1 and object 2 (object 1 - object 2) for scenario 1. During the first period, both models classify the views of object 1 correctly except for the HMAX model at the end of the first period (where the blue line is negative). For the second period, the HMAX model misclassifies the views of object 2 most of the time. In contrast, the RBF model performs correctly most of the time.

both models.

In figure 5.14A the model activity is presented for scenario 2. The presence of noise in the centroids of the blobs detected plays an important role in the performance of the models. In this scenario, the recognition signals for each object are more similar to one another in contrast to scenario 1. Therefore, the test views are more difficult to classify for this scenario (compare the differences between the recognition signals for both objects in figures 5.13B and 5.14B). This confirms the predictions made in the previous section where it was shown that the presence of noise in scenario 2 would increase the similarity of views processed by the models, particularly for the HMAX model.

In scenario 3, 16 views are used during training and no noise is considered. In this case, the recognition signals of object 1 and object 2 are significantly distinct. As suggested in the previous section, views of object 1 are easier to recognise (since the similarity amongst views of object 1 is increased) in period 1 for both models. However, in period 2 only the RBF model can clearly distinguish between the recognition signals of both objects (see figure 5.15B).

In scenario 4 (16 views and noise), as with the results in the previous section, the similarity between views of the same object increases when more views are added (especially for object 1). However, when noise is added to the BDM, the similarity increases (see figure 5.12). When the models are tested using trajectory 1 and trained in scenario 4, the recognition signals of the models are different even when noise is present for period 1 (compare the first 55 steps in figures 5.16A and 5.14A). However, when object 2 is within the field of view (period 2), the signals become similar for the RBF model and the HMAX model.

After observing the results for the recognition task for the RBF and HMAX model

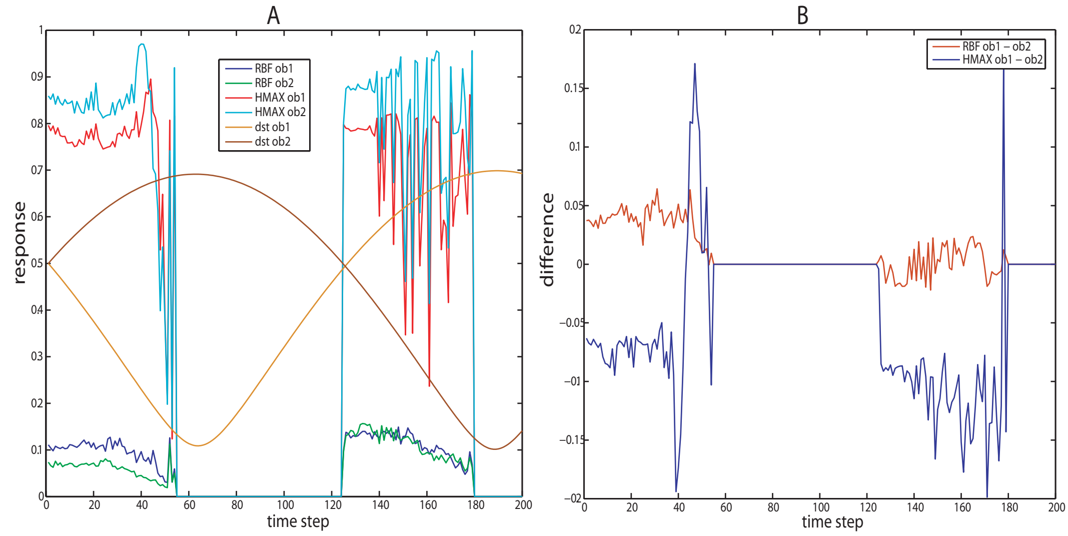

Figure 5.14: (A) Recognition signals of the two models for trajectory 1 for scenario 2. The signals of both objects become more similar, compared to the signals in scenario 1. (B) Difference between the recognition signal of object 1 and object 2 (object 1 - object 2) for scenario 2. During the first period, the RBF model classifies the test views of object 1 correctly (red line is positive). In contrast, the HMAX model misclassfies them most of the time (blue line negative). For the second period, the RBF misclassifies the test views of object 2 (positive red line), while the HMAX model classifies them correctly (negative blue line).

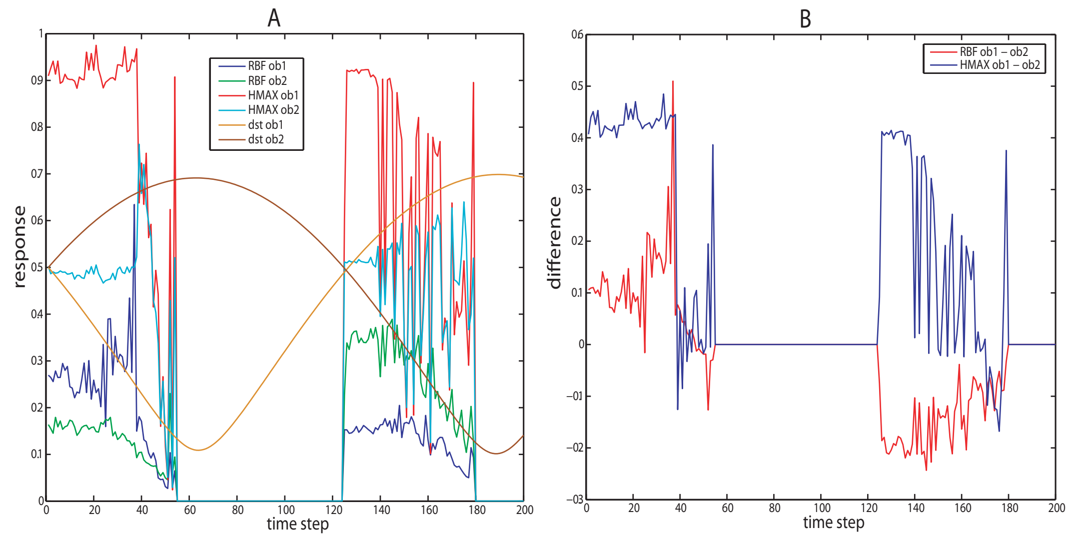

Figure 5.15: (A) Recognition signals of the two models for trajectory 1 for scenario 3. With the absence of noise in the blob detection mechanism and more training views, the difference between the recognition signals is larger than in the previous cases. (B) Difference between the recognition signal of object 1 and object 2 (object 1 - object 2) for scenario 4. During the first period, the recognition signal for object 1 is higher than the signal for object 2 for both models. However, for the second period, only the RBF signal for object 2 is higher than the signal for object 2.

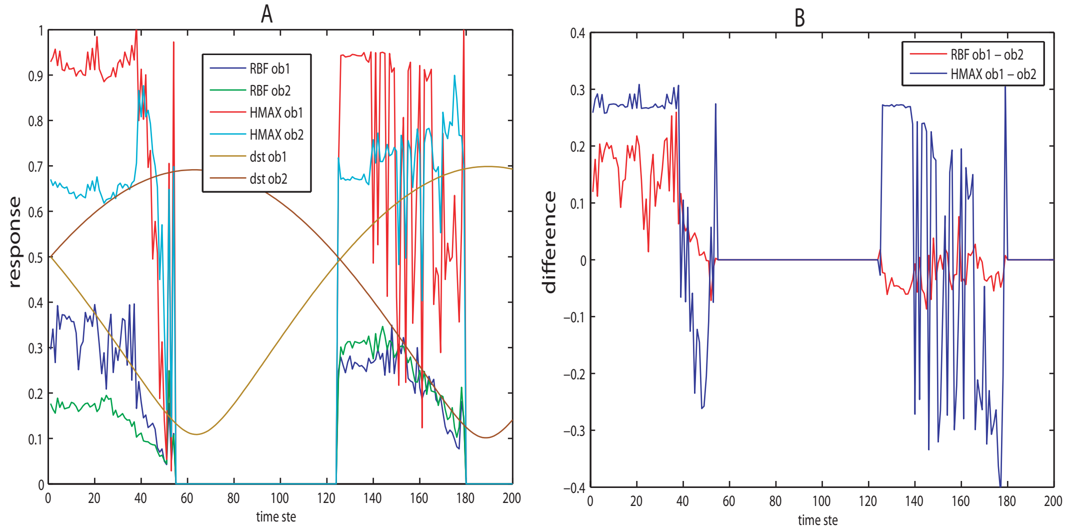

Figure 5.16: (A) Recognition signals of the two models for trajectory 1 for scenario 4. (B) Difference between the recognition signal of object 1 and object 2 (object 1 - object 2) for scenario 4. During the first period of the test, both models classify the test views correctly, except at the end of the first period for the HMAX model (negative blue line). For the second period, the HMAX model becomes chaotic. The RBF model classifies the test views most of the time.

trained using the four scenarios and tested in trajectory 1, the general case can be summarised as follows. When less views are used during training (scenarios 1 and 2), the similarity maps show a significant sensitivity to noise in the BDM, especially for the RBF model (RBF signals for object 1 and 2 are very similar when noise is added to the BDM). When more views are used during training (scenarios 3 and 4), the RBF and HMAX model show an increase in robustness to noise (in comparison with scenarios 1 and 2). In contrast, the HMAX is less affected by noise with the increase in the number of views (given that the HMAX views are very similar, the views become easier to discriminate by adding noise, especially for object 2). Another important thing to notice is that once the agent is close to the object, the separability of the test views decreases dramatically, this case was not predicted by the similarity maps given that the distance between the object and agent when the training views were collected was constant.

Similarity maps for the test phase. In order to have a better idea about the similarity of the views between the test and the training views, the similarity maps during the test for each scenario are presented in figure 5.17.

Figure 5.17 shows how similar the current view is at every time step in the test trial with respect to the training views. Even when the agent is at a distance similar to training, the view can be very different to the ones used in training. Blue regions in the maps represent a high similarity (small distance) between the current view during the test and the corresponding training views at every time step in the trial. Conversely, the red regions represent low similarity (large distance) between the current test view and the corresponding training view. So for period 1 of trajectory 1 (test), the lower views should be bluer (lower left). Analogously, for period 2, the right upper region of the map should be darker than the right lower region.

The similarity map for the test phase confirms the predicted overall models' behaviour

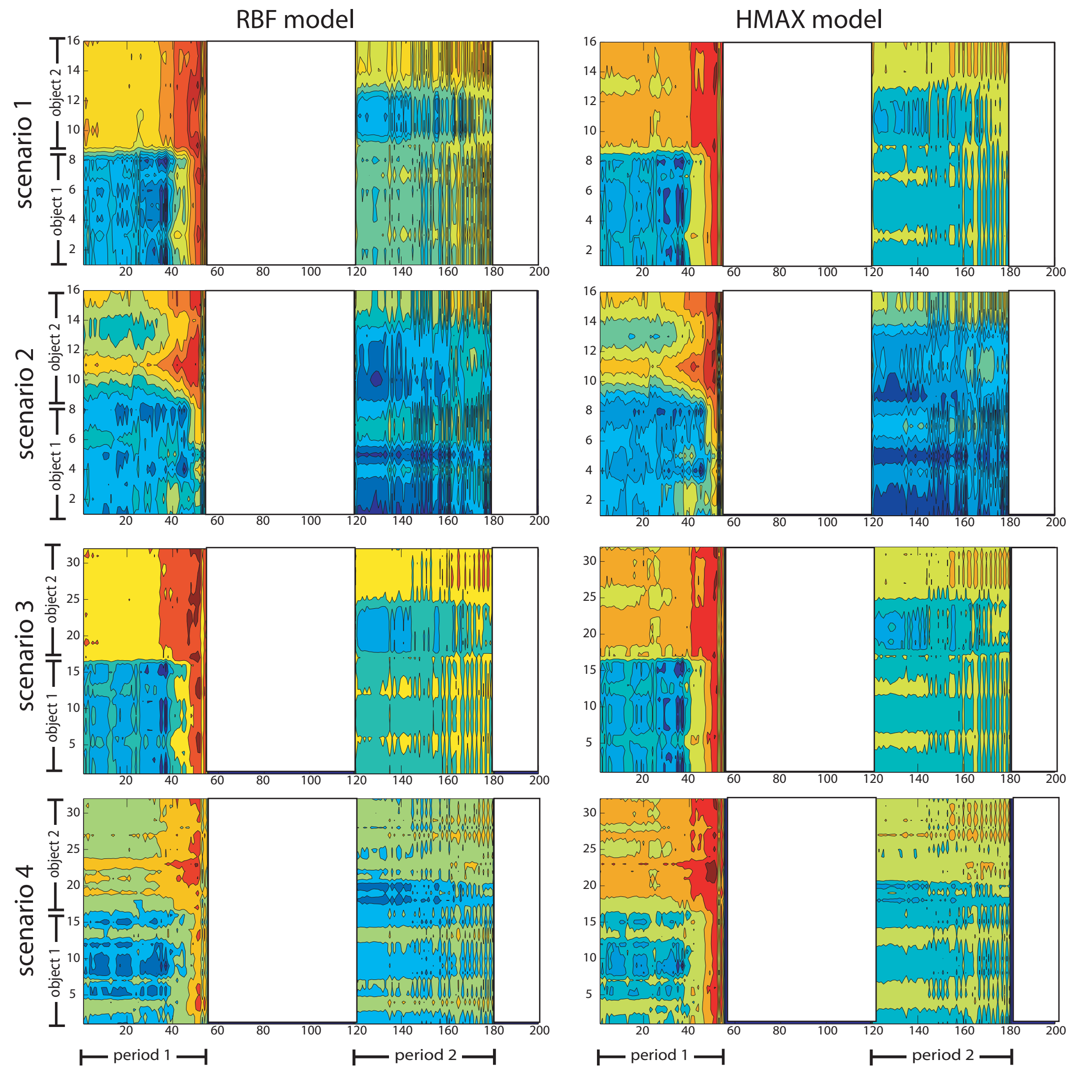

Figure 5.17: Similarity maps for the test phase when the agent is following trajectory 1. The left column represents the maps for the RBF model and the right column, the maps for the HMAX model. The different rows correspond to the scenarios in which the models were trained.

by the activity model figures. For scenarios without noise (1 and 3), the RBF similarity maps (left column) clearly show a dark blue region in the lower left area (corresponding to period 1) and a darker region in the upper right area (corresponding to period 2). For scenarios with noise (2 and 4) these maps show that in the case of using more views during training (scenario 4), there is a clearer distinction of the right blue regions in the right areas. Additionally, these maps clearly show the lower discrimination capability of the HMAX model especially for object 2.

In general, we can see that the noise introduced during training affects the performance of the RBF model but this effect is decreased when more views are added to the training. In contrast, the performance of the HMAX model was not significantly changed in the different scenarios for trajectory 1.

Testing using trajectory 2

When using trajectory 1 during testing, the models relied on the points of view that the models were exposed during training, to account for the variation in pose of the objects during testing. I demonstrated that in the case that the views are processed by the RBF model, the discriminability increases. In contrast, when the views are processed by the HMAX model, the discriminability of the views decreases. In this section, I analyse the case when only one point of view is provided during testing.

For trajectory 2, the point of view is kept constant but the scale is changing. Therefore, in this case the models are tested against scale invariance but not against rotation invariance. In figure 5.18, the activity of the models is presented for each scenario and for each object when trajectory 2 is used during the testing phase. The left column corresponds to the model activity when object 1 was in the field of view and the right column corresponds to the model activity when object 2 was in the field of view. The rows show the different model activity when the models were trained in the four scenarios.

When the models were trained with 8 views in scenarios 1 and 2 (rows 1 and 2), the models were significantly affected by noise. In scenario 1, there is a peak in the activity of the RBF model which corresponds to the time the current view closely matches one of the training views. This peak does not exist when there is noise added to the BDM (scenario 2) because a close match is not possible since the centroid of the blob detected is randomly shifted. This shows the high specificity of the RBF model. In contrast, the activity of the HMAX model shows a higher activity spread around the area where the distance is close to the training distance. When more views are used during training in scenarios 3 and 4 (rows 3 and 4), the RBF model shows a significant increase in the robustness to noise for both objects (left and right columns). However, the HMAX model only shows some degree of increase in the robustness to noise in the case of object 1 (left column).

5.4 Discussion

The experiments in this chapter show that the similarity of the views being processed by the models was modified differently by each model. For the RBF model the similarity between views of the same object was increased compared with the image views, additionally, the similarity between RBF views of different objects was decreased. In contrast, the

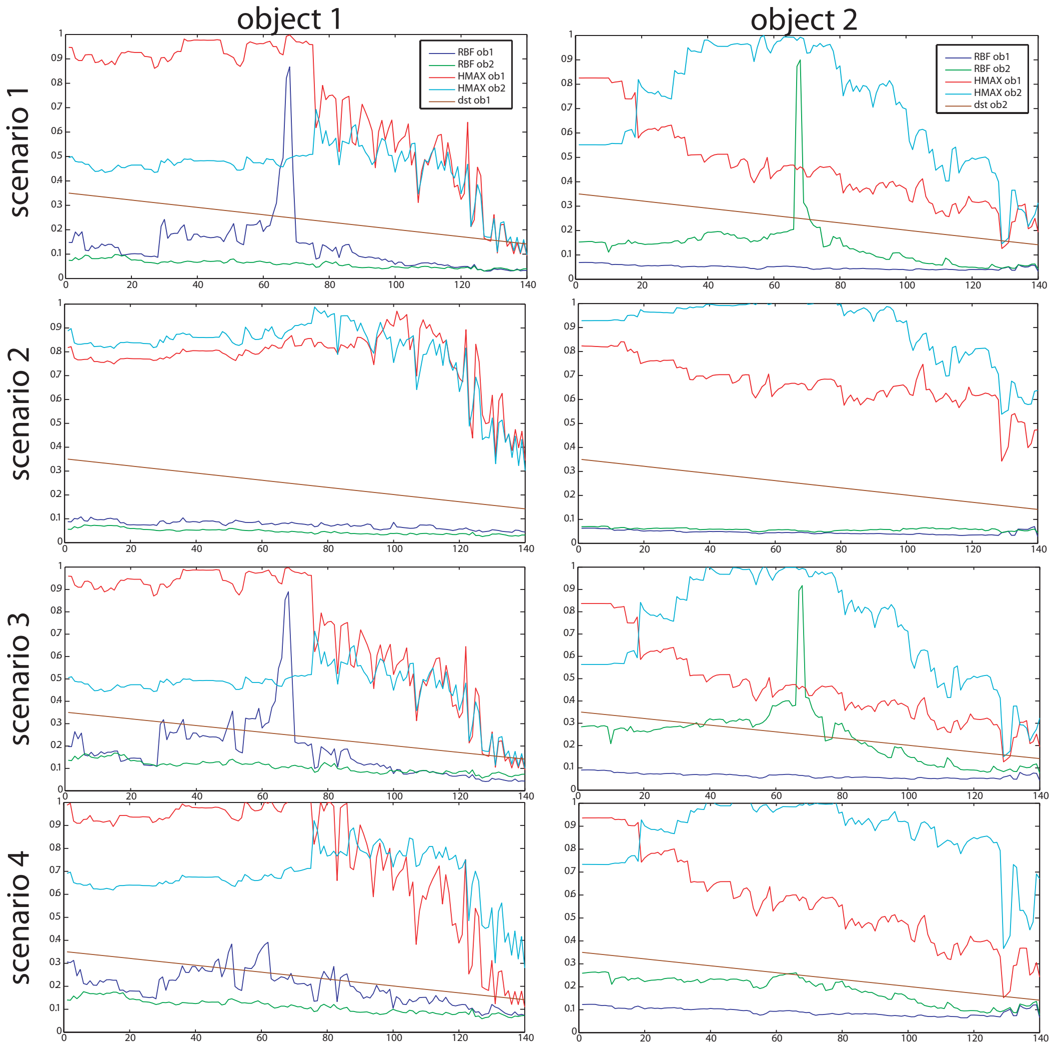

Figure 5.18: Model activity during the testing phase when the agent was following trajectory 2 for the four scenarios. Given that the output of the models is deterministic, the model activity shown in this figure is the same every time the agent approaches the object in the same way.

similarity was increased for HMAX views between both, the same object, and different objects. Additionally, it was demonstrated that when using more views, the RBF model shows a higher robustness to noise in the BDM. In this section I discuss why the similarities between the views are altered differently after being processed by the RBF and HMAX models.

5.4.1 Dimensionality

The first reason for the difference of the similarity between the RBF views and HMAX views was the dimensionality of the output of the models. The dimensionality of the input is 4800, given by the size of the images (60×80 pixels), whereas the dimensionality of the output of the RBF model is 76800, given by the number of orientations and the number of the filters employed (rows $\times$ cols $\times num_f \times num_o = 60 \times 80 \times 4 \times 4$ where num_f is the number of filter sizes, num_o is the number of filter orientations). The dimensionality of the output of the HMAX model is the size of the last layer of the model (C2), which was 256. Therefore, the dimensionality of the output of the RBF is significantly larger than the dimensionality of its input. In contrast, the dimensionality of the output of the HMAX model is significantly smaller than the dimensionality of its input.

The difference in the dimensionality of the output of the models has an important role in the discrimination capability of the models. When the output of the model has high dimensionality, the discrimination power (specificity) increases since the description of each element becomes increasingly more complete (specific). However, the generalisation power decreases since the template used to recognise an element is more specific. In contrast, when the output of the model has low dimensionality, the discrimination power decreases as the description of each element becomes increasingly more general (less specific). However, the generalisation power increases since the template used to recognise an element is more general. The difference in the generalisation and specificity of the models plays an important role in the separability of the objects being analysed. In the case of RBF, the template of the objects is expanded (with the increase in the dimensionality) and in the case of the HMAX, the template of objects is collapsed (due to the reduction of dimensionality).

5.4.2 The role of the BDM

The second reason for the difference in the similarity between RBF views and HMAX views is that, given the specificity and generalisation, the BDM has an important role. Since the generalisation of the HMAX is high, particularly for translation and scale (Riesenhuber and Poggio, 1999b), the BDM does not add much to the model. However, in the case of the RBF, the high specificity allows this model to take advantage of the translation and scale invariances provided by the BDM. This translates into an increase in the discrimination power of the RBF model and a significant degree of generalisation when provided with the BDM.

Adding noise to the centres of the blobs detected significantly impacted the RBF model compared to the impact it had on the HMAX for both trajectories (see figures 5.17 and

5.18). This perturbation of the RBF model, when fewer views were used during training, was due to the high specificity of this model, compared to the high generalisation of the HMAX model.

To summarise, the performance of the RBF model relies on whether or not the test views are similar enough to the training views (due to the high specificity of the model). In contrast, the performance of the HMAX model relies on a less specific presentation of views (due to the high generalisation of the model). In the case where the models were tested using trajectory 1, the increase in the robustness to noise by the RBF when more views are considered (scenarios 2 and 4) is due to the increase of the similarity between the RBF views of the same object and the decrease of the similarity between the RBF views of different objects (due to the specificity of the model). The HMAX model is not significantly affected by the noise in the BDM because the similarity between HMAX views was already high (due to the reduction of the dimensionality and the high generalisation). Similarly, by adding more views, the similarity between the HMAX views was only increased.

When the models were tested using trajectory 2, the specificity of the RBF model is evident when no noise is present (scenarios 1 and 3). In this case, there is a peak in the activity of the model in figure 5.18 which corresponds to a close match between the test view and the training view when the distance between the agent and the object during testing is similar to the distance used during training. This peak does not exist in scenarios 2 and 4 since a close match is not possible (as noise is present). However, the difference between RBF signals in scenarios 2 and 4 demonstrates the increase in the robustness to noise in the BDM. The HMAX model activity when using trajectory 2 during testing demonstrates that when the point of view is fixed and only the distance is changing, the HMAX model performs better than when the point of view is changed during testing.

The difference between the RBF model activity when the models are tested using trajectory 1 and trajectory 2 suggests that movement can affect the recognition performance of the models (see the difference between the RBF signals in both trajectories in figures 5.13-5.16 and figure 5.18 especially for object 2).

5.5 Conclusion

In the previous chapter it was demonstrated that it was not possible to use an ER approach to find controllers that would provide the active vision characteristics necessary in order to employ a simple model of object recognition with desirable 3D transformation invariances. However, another implication of embodiment was left unexplored. If simpler movements are necessary to improve the recognition performance of a simple model, the required controllers could be simpler. The first step to discovering whether or not this is a viable option is to investigate whether or not movement can be exploited by the HMAX and RBF models to improve object recognition.

Therefore, in this chapter I explore whether movement can impact the performance of the models by training the models using views collected when the agent was following a circular trajectory and tested when the agent was following one of two trajectories to approach the objects. In the first case, the objects were approached by the agent following an arc trajectory. In the second case, the agent approached the objects following a straight line trajectory. In order to evaluate whether the models could exploit movement to overcome variations to different conditions during training, the number of views was increased from 8 to 16 and random noise was added to the BDM.

The results in this chapter demonstrate that by having a large enough number of views during training, the RBF model can be robust to noise in the visual system. Additionally, it the difference in the activity of the RBF and HMAX models when tested in different trajectories suggests that there could be certain movement that would be better in the recognition process than others. This suggests that given that the RBF model can be robust to noise in the visual system and improve the recognition performance by exploiting movement, there is a good possibility of simplifying the complexity of the controllers required to perform object recognition in an autonomous mobile agent using a simple biologically inspired model. However, the question of what movement strategies are best and how they can be exploited by the RBF is left unexplored. This issue will be investigated in the next chapter.

Chapter 6

Movement strategies during learning

6.1 Introduction

Mobile visual systems exploit visual regularities as they gather information by moving, not only in space but also in time (Chen and Chen, 2004; Aloimonos, 1993). In active vision, the exploitation of movement is an important issue. The exploitation of movement can be beneficial for several reasons, for example, movement can provide access to multiple points of view of an object that would help to discriminate objects. Also, the variation in visual information imposed by movement, not only in space but in time, can provide advantages to mobile visual systems. An important aspect to evaluate in the exploitation of such variations in the visual information is its robustness to perturbations. If the exploitation of such variations is robust to perturbations or noise in the visual system, then this approach can be employed in visual systems that deal with significant variations in environmental conditions, such as mobile agents.

In the previous chapter the HMAX and RBF models were tested using two different trajectories in order to find out whether models exploit the variation in the visual information provided by movement. The results suggest that the variations in incoming visual information are exploited in a different way by each model. In this chapter, four experiments are carried out to further investigate how visual information varies with simple movement strategies and how the models exploit these variation. In the first experiment I study the way the models exploit the variation in the visual information through four different movement strategies. For the next three experiments, only the RBF model is considered since the results from the previous chapters indicate that this model outperforms the HMAX model when enough rotation and scale invariance is provided during training by the movement strategies. In the second experiment I present another way of exploiting movement for recognition. This consists of using temporal information which is represented by the difference between consecutive views (DBCV) instead of using single view presentations (SVP). In the third experiment the robustness of the RBF model to perturbations to the trajectories using the DBCV is presented. Finally, in the fourth experiment I analyse the case when movement has to be employed to acquire multiple views in order to be able to discriminate similar objects. Additionally, the recognition performance of the model is analysed in the case that more objects are added to the system.

These results suggest that exploiting the dynamics of agent-environment interaction can, in certain circumstances, obviate the need for complex models of visual object recognition. They also show that the performance of the RBF model can be enhanced by training and testing using dynamic visual signals generated during each movement strategy. Given the properties the RBF model demonstrated in this chapter, it can be considered a good candidate for the implementation of a simple model of object recognition in an autonomous agent in the real world.

6.2 Methods

The general setup in the four experiments is the same as in the previous chapter, involving a simulated agent performing a simple object recognition task. The agent-environment system comprises a simple wheeled agent in a flat planar environment containing two objects (a 'kettle' and a 'bolt'), simulated using the OpenGL library (see figure 6.1). The visual object recognition system of the agent comprises three parts: a 'blob detection mechanism' (BDM), an 'analysis module' consisting of either the HMAX or the RBF model, and a 'classifier module' which classifies the output of the analysis module into one of two categories ('kettle' or 'bolt'). Each experiment consisted of two phases. First, a learning phase in which the agent followed one of four different movement strategies (see figure 6.2) while collecting training views which are used to train either the HMAX or the RBF model. Second, a testing phase, during which the agent follows a separate movement strategy (testing trajectory) while collecting views used to test object recognition performance.

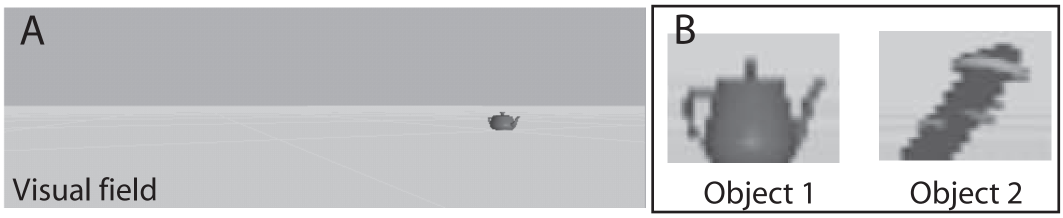

Figure 6.1: (A) Visual field of the agent: shows object 1 in the field of view. (B) Sample views of object 1 and object 2: object 1 is a rounded object so it does not have a significant variability with respect to rotation, in contrast, object 2 has a significantly higher variability with respect to rotation due to its vertical inclination.

In addition to the general set up, there are specific variations for experiments 2 through 4. For experiments 2 and 3, the input of the classifier module is not only a single view presentation (SVP), but also the difference between consecutive views (DBCV) to represent temporal information (definition below). For experiment 4, apart from the initial two objects, four objects were considered. A description of these objects will be given when this experiment is presented.

Blob detection mechanism. As mentioned in the previous chapter, the BDM selects the area of the visual field containing the object (see details of the BDM in section 3.2.3).

Cropped regions returned by the BDM are normalised to 60 × 80 pixels (a blob) before being processed by the analysis module.

Analysis module. The analysis module is also the same as used in the previous chapter. After the regions are selected by the BDM, the analysis module, which can be either the RBF or the HMAX model, processes the visual information. The details of each model can be revisited in chapter 2 and 3.

Classifier module. Again, the classifier module employed is the same one presented in the previous chapters. It consists of a RBFN with view tuned units centred in each view of the objects (see chapter 3 for details).

Movement strategies The training views were collected when the agent was following one of four different trajectories (these trajectories are called movement strategies throughout the rest of this chapter).

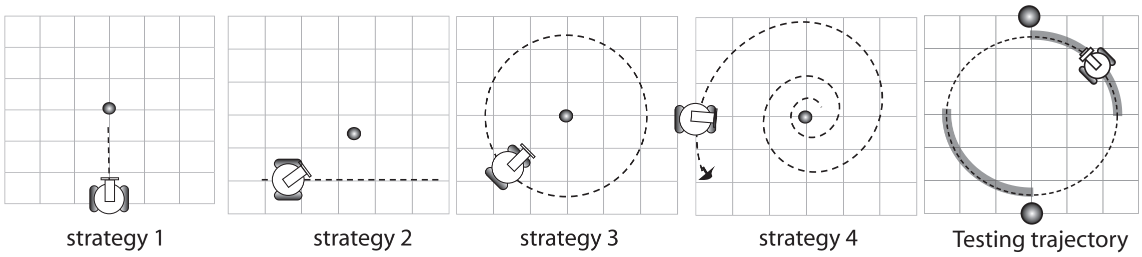

Figure 6.2: Movement strategies. While following the movement strategies, the agent takes snapshots at uniform intervals. Strategy 1: the agent approaches the object in a straight line. Strategy 2: the agent passes the object following a straight line. Strategy 3: the agent circles the object with a fixed radius. Strategy 4: the agent spirals the object. The testing trajectory consisted of two phases which correspond to the grey segments. In the first period object 1 was within the field of view and, in period 2 object 2 was within the field of view.

The properties of the set of training views changed depending upon the movement strategy used during their collection. These strategies were designed in order to provide different properties in the training views (see figure 6.2). Movement strategy 1, for example, allows the agent to exploit the different training distances while using the same point of view. Therefore, the training views using this strategy only provide variance in scale. Strategy 2 provides a small degree of variance in perceived rotation (points of view) and a small degree of variance in scale as well, since the agent is passing in front of the target object. The point of view changes slightly as the distance between the agent and the object changes. Strategy 3 provides only variance in points of view since the distance between the agent and the object is always the same, while the point of view changes for each training view. Strategy 4 provides a combination of variance in scale and point of view since the distance and the perspective of the agent to the object are changing continuously. For each strategy, 16 training views are taken for each object at regular time intervals. Therefore, training phases varied in length from 160 to 200 time steps depending on the movement strategy used.

In the testing phase, the agents followed a trajectory (testing trajectory) that differs from the movement strategies used in the learning phase. The testing trajectory was designed so it would resemble a plausible situation in the real world where the objects are approached in a natural way that provides views of the objects from multiple angles and scales (see figure 6.2). The testing phase lasted for 200 time steps. During the first 55 steps (period 1) object 1 was present in the visual field and during 125-180 (period 2) object 2 was present in the visual field.

Temporal information. In experiments 2 and 3 of this chapter an exploration of temporal exploitation through movement is considered by taking the absolute difference between consecutive views (referred to as DBCV in the rest of this chapter). The difference $DBCV$ between consecutive views i and j, is calculated as

$$DBCV(i,j) = 1/2 \cdot |RBF(i) - RBF(j)| \tag{6.1}$$

That is, the absolute difference is taken after the views have been processed by the RBF model (DBC has the same dimensionality than $RBF(i)$). Otherwise, when no temporal information was considered, only one view was presented to the model. This case is referred to as single view presentation (SVP).

6.3 Experiment 1: Movement strategies

In the previous chapter it was suggested that some movement strategies could be better than others for improving the recognition performance of the object recognition models. In this experiment I investigate this by using four movement strategies to collect the training views. To assess the recognition of the models for each movement strategy, the RBF and HMAX are first trained using the different movement strategies shown in figure 6.2. After that, the models are tested while the agent traverses the testing trajectory shown in figure 6.2. The performance of the models being trained using the different movement strategies during learning is shown in figure 6.3.

For strategy 1, HMAX outperforms the RBF model. Since this movement strategy presents the objects from a single point of view, the models can only acquire scale invariance. For a simple model like the RBF, this strategy would only work if the objects in the testing phase were viewed from a similar perspective to the one from training phase. Since this is not the case (the point of view is changing and is different from training), the RBF model cannot closely match test views to training. However, the HMAX model is able to generalise when a limited point of view is provided during training. This is because the features extracted by the HMAX from a single perspective captures higher order properties of the objects which are in some sense independent of the angle it is viewed at. For strategy 2, the results are very similar to strategy 1 as the training views are again taken from a limited set of angular positions. However, when the point of view is varied significantly during the training phase as in strategy 3, the RBF model's performance increases greatly. Since the number of points of view is significantly increased, the RBF can achieve a close match between the training and the test views. In contrast, the HMAX's performance decreases, demonstrating that its discriminability can be reduced when the variability of the training views is increased. Similar results are obtained for strategy 4 where both point of view and scale are changed during training.

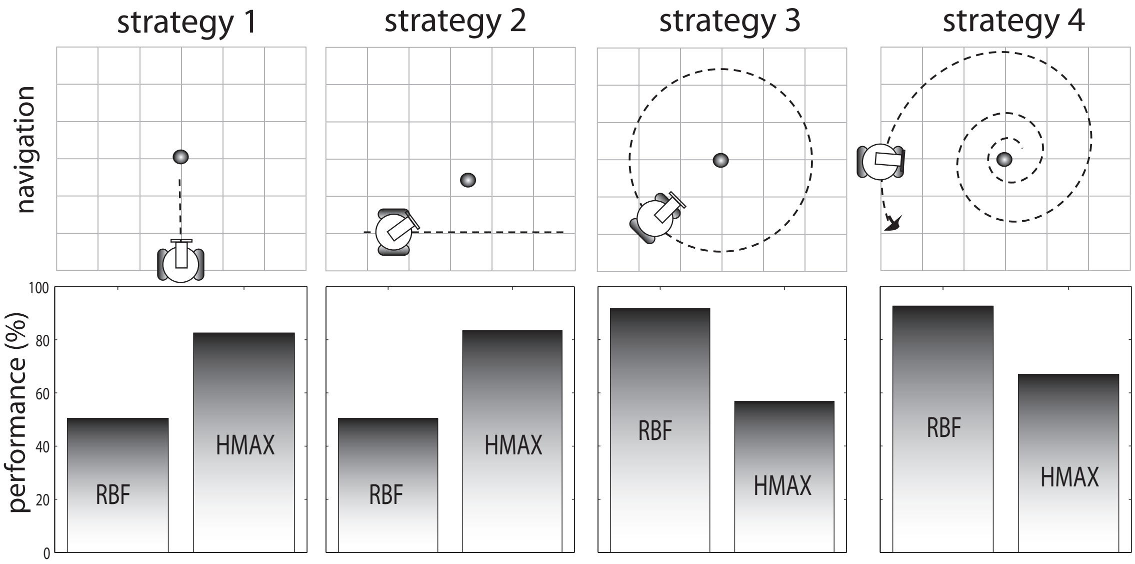

Figure 6.3: Movement strategies and model performance. The performance of the RBF model increases when the movement strategies allow it to exploit the rotational information during training. In contrast, the HMAX model performance decreases when the model is exposed to multiple rotational views during training in strategies 3 and 4. The performance of the models refers to the number of times the model has a correct guess over the test phase (only averaged over the total number of presentations during the test phase). Since this number depends on the presentation of object views which are deterministic (the same views will be presented every time the agent follows the corresponding trajectory) and the input-output mapping of the models is deterministic, the bars in this figure do not consider any statistical measure of variance. Note that chance level is 50%.

I now explain why the training strategy determines the performance of the models. The reason for a performance change with different movement strategies has to do with the way the objects change with the movement of the agent and also with the features detected by each model. The objects used in this experiments are the teapot (object 1) and the bolt (object 2) used in the previous chapter (figure 6.1). In particular, the variability of the objects to rotation is significantly different. As object 1 is sphere-shaped, its image does not change significantly when the agent rotates around it (especially at large distances). In contrast, object 2 has an off vertical orientation which makes it variable when the point of view is changed.

Since the RBF model responds mainly to oriented edges, its response depends on a close match between the test and the training views and we would expect it to fail when a close match is not possible. When the points of view are limited (strategy 1 and strategy 2), a close match between the training and testing views is not possible (since the perspective in which the agent approaches objects is different for strategies 1 and 2, and the testing movement strategy). In particular, object 2 is difficult to discriminate, as it changes significantly along the testing trajectory. Thus the overall performance on these strategies is around 50% (see performance for movement strategies 1 and 2 in figure 6.3). Because the HMAX model acts on a combination of the dominant features detected by the RBF (since its first layer is the RBF), it responds to a more generalised pattern of features, rather than a close match. Since object 1 does not change significantly, the dominant features will be the ones responding to the main orientation of the object (horizontal). For object 2, if the object is seen from a single point of view, the dominant features will be the ones corresponding to the main orientation of the object, roughly 30 degrees from the vertical in the case of strategy 1. These features (which form the HMAX template for object 2), will be different to the dominant features detected for object 1 (which form the HMAX template for object 1), so the discriminability of the HMAX model is high in this case (figure 6.3).

In contrast, when the point of view is varied significantly during the training phase (strategies 3 and 4), the RBF achieves a close match between training and testing views. Since there are more points of view in the training set, the model can cope with object rotation. In the case of the HMAX model, since object 2 changes its orientation during training, the model extracts dominant features in many orientations, which form a very general template and thus decrease object discriminability. This scenario is depicted in figure 6.4 which shows the models' output after training with strategy 3. The objects are within the field of view in different periods (grey segments in figure 6.2) during the 200 time step trial. In period 1 (1-55 time steps) object 1 is within the field of view, and in period 2 (125-180 time steps) object 2 is within the field of view. For the RBF model, the agent can correctly discriminate both objects. Note the peak in output that corresponds to a close match between test and training view (around time step 37). In the case of the HMAX model, while there is no problem with period 1, in period 2 discriminability is reduced significantly.

Similarity maps further explain the discrimination ability of the models (figure 6.5).

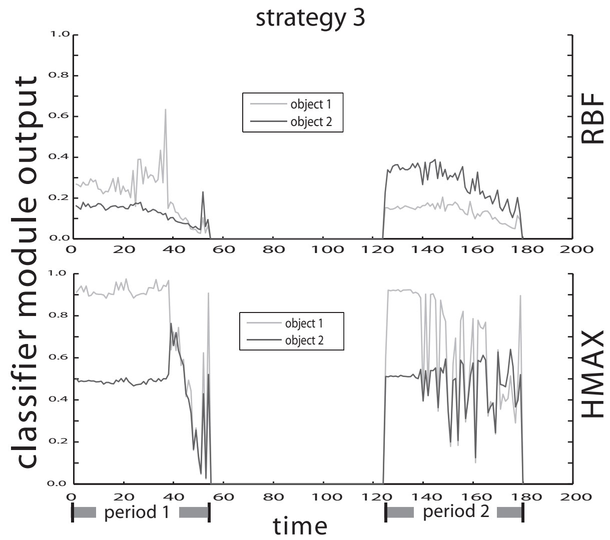

Figure 6.4: RBF and HMAX models activity during the test phase using strategy 3. When the movement strategy provides multiple points of view during the learning phase, the RBF can have a close match between the training and the test views. In contrast, the HMAX model decreases its discriminability when more points of view are considered. Period 1 represents the time when object 1 is within the visual field. Period 2 is the time when object 2 is within the visual field.

As described in the previous chapter, a similarity map is a diagram representing the similarity between the current view and the training views of the objects (Y axis) at every time step (X axis). Every point in the map has a grey-scale value dependent on the distance between the current view and the training view after processing by the analysis module. The darker a point, the smaller the distance between the views, where distance is the sum of the absolute difference between the views. Each map is divided in two periods which correspond to points where the objects are in the agents' visual field (see figure 5.2A). In the first 55 time steps (period 1), object 1 is present in the visual field and during period 2 (from 125-180), object 2 is in the visual field.

The upper part of figure 6.5 shows the similarity between views for the RBF, while the lower shows the similarity map for the HMAX model (HMAX views). If a model was responding correctly, we would expect darker areas in the lower region of the similarity map for period 1 and in the upper region of the similarity map for period 2. The similarity map for the RBF has these general features as it has acquired a degree of both rotation and scale invariance from the training trajectory. The responses of the HMAX model however, show that the higher level features extracted for each object are too similar for the two objects to be discriminated reliably.

Thus we see that the performance of the HMAX model is higher than the performance of the RBF model when the rotational variation of the training views is low (strategies 1 and 2). In contrast, when the movement strategies provide high variation in the training views (strategies 3 and 4), the performance of the RBF model is higher than the performance of the HMAX model.

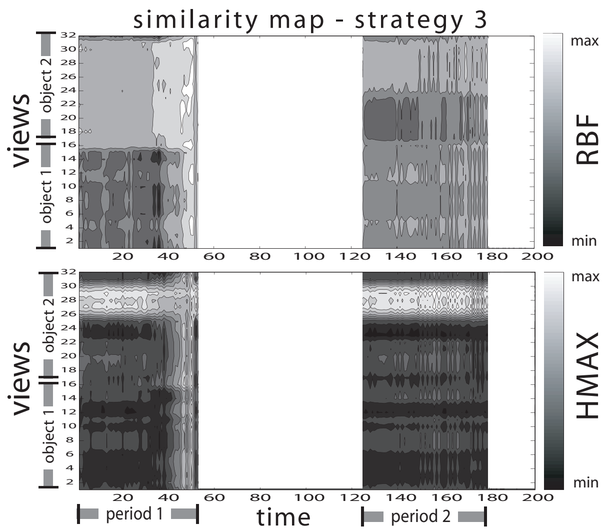

Figure 6.5: Similarity maps of the models using strategy 3. The darker the regions in each map, the more similar the corresponding views. For the RBF map there is an obvious darker region in the left lower area (corresponding to the views of object 1) for the first period, and a smaller darker region in the right upper area (corresponding to the views of the object 2). In contrast, for the HMAX similarity map dark areas appear during both periods for views associated with both objects.

6.4 Experiment 2: Temporal information using the RBF model

In the previous experiment it was shown through four different movement strategies how the two models exploit multiple viewpoints in the visual acquisition process. In particular, when enough variations and number of views are provided during training, the RBF model outperforms the HMAX model. Therefore, since these are the conditions that will be provided for the rest of this chapter, from now onwards I will focus solely on the RBF model.

In this experiment I investigate whether temporal information can be exploited by the RBF model. The temporal information is represented in this experiment as the absolute difference between consecutive views (DBCV) after being processed by the model. That is,

$$DBCV(i,j) = 1/2 \cdot |RBF(i) - RBF(j)| \tag{6.2}$$

where i, j are object views and $RBF(i)$ is the output of the RBF model after processing the view i.

First, I evaluate the performance of the RBF model using DBCV to find out whether or not this type of structure can be used for object recognition. After that, I investigate whether or not the temporal structure of the DBCV is exploited by the RBF model.

Model performance using DBCV. The RBF model was trained using the four movement strategies and evaluated using the testing strategy during the testing phase. This was the same setup as when a single view was presented to the system (SVP) in the previous experiment, but this time using the DBCV.

A comparison of the performance of the model when using SVP and DBCV is presented in table 6.1. The results are broadly similar to the case when a single view (SVP) is presented to the model showing that DBCV can be exploited by the RBF and provide the same invariances to rotation and scale as when the model was trained using SVP.

strategy	SVP	DBCV
1	50	51
2	51	56
3	92	73
4	94	95

Table 6.1: Comparison of the performance (%) of the RBF model using SVP and DBCV using the four movement strategies during training and the test trajectory during the testing phase. The performance refers to the number of times the models guess correctly over the number of time steps in the test phase.

Exploitation of the temporal structure in the DBCV.

In order to test whether or not the RBF is exploiting the temporal structure of the DBCV, the model was trained in two different conditions. In the first one, the order of the training views was the same order in which the views were collected while the agent traversed the corresponding training strategy (normal order). In the second condition, the order of the training views was randomised before being processed by the model (random order). That is, if during training the views were $v_1, v_2, \ldots, v_{16}$ (normal order), in the second condition the order of the views to calculate the DBCV could be $v_{15}, v_8, v_1, \ldots, v_2$ for example.

A significant difference in the model activity (ie output of the classification module) between the cases when normal order and random order of the training views are employed means the model activity depends on the order the views were presented during training, which in turn demonstrates an exploitation of the temporal structure in the DBCV by the RBF model.

Since the exploitation of temporal structure depends on the changes the objects' views undergo while the agent is moving, the most suitable movement strategies to use in this experiment are the ones that provide multiple points of view and scale variation. The movement strategies that provide such types of variances are strategies 3 and 4. Therefore, I will use only these strategies when analysing the temporal information.

Additionally, I will mainly focus on object 2 because this object has an off vertical orientation, therefore, it is significantly variable in rotation. In contrast, object 1 is a rounded object and so appears similar from any perspective (so the order in which the views of object 1 are presented is not significantly relevant).

Figure 6.6 shows the RBF recognition signal (model activity) using DBCV when strategy 3 was used during training and testing. When the order of the training views is randomised (in the sense that the training views are presented to the system in a random order, rather than in the sequential order in which the training views were originally collected), the recognition signal in the model drops (right column) in comparison with the case when the training views were presented to the model in the normal order (left column). This shows that the order in which the views are presented to the model is important, therefore, when the training and testing trajectories are similar, the RBF model exploits the temporal structure of the DBCV.

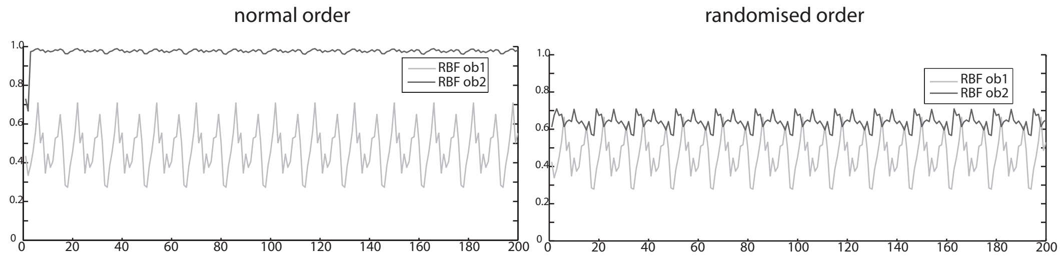

Figure 6.6: RBF recognition signal (model activity) using the DBCV and trained using strategy 3 and tested in the same trajectory with randomised and normal ordered training views. The left column figure shows the RBF recognition signal for normal conditions and the right column figure shows the recognition signal for random ordered training views.

In order to test the exploitation of the temporal structure in the case where the trajectory used during training and testing are different, the RBF model was trained using strategy 4 and tested using the testing trajectory. Once again, two cases are presented, one when the order of training views is normal and the other one when the order of the training views is randomised.

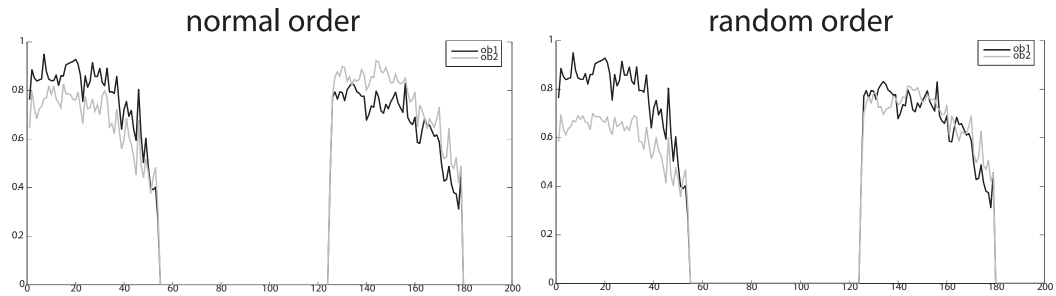

Figure 6.7: RBF model activity when trained using strategy 4 and tested using the testing trajectory. Left column: normal order of the training views. Right column: random order of the training views.

Figure 6.7 shows the model activity in the case when strategy 4 was used during training and the testing trajectory during testing. In this case, the exploitation of the temporal structure of consecutive views is less distinctive since the difference between the model activity (when object 2 is present in the visual field) between the ordered and randomised case is very small especially for period 2. [1]

This experiment shows that the exploitation of the temporal structure is affected by the way the visual information is presented to the model, meaning that the RBF model can exploit the temporal information provided by the DBCV. In the next experiment a study of the robustness to perturbations to the training and testing trajectories for the RBF model using temporal information is presented.

[1] Different randomised orders of the training views were tested (data not shown) and for particular randomised orders the difference between the activity of the model when the order was normal versus when it was randomised, was almost nonexistent.

6.5 Experiment 3: Robustness of the RBF when using temporal information

The previous experiment demonstrated that the exploitation of the temporal structure by the RBF model using DBCV depends on how similar the variations in the views are (imposed by the trajectories) during the training and testing. In this experiment I further investigate this issue by analysing the exploitation of the temporal structure when the training and testing trajectories are perturbed.

First I analyse the robustness of the RBF model when using the DBCV to perturbations of strategy 3 during testing. The perturbations to strategy 3 consisted of varying its radius or moving its centre. After that, I investigate the robustness of the RBF model when using DBCV to perturbations of strategy 4. The perturbations to this strategy consisted of moving its centre and varying the intervals when the views are taken.

6.5.1 Changing the radius of strategy 3

The first robustness test consisted of increasing the radius of the trajectory used during testing. In the following figures the activity of the RBF model is presented using strategy 3 during training and modifications of the same trajectory during testing while using object 2 (chosen for its variability in consecutive views, as in experiment 2). The activity of the model is shown using normal and random order of the training views presented to the model using DBCV.

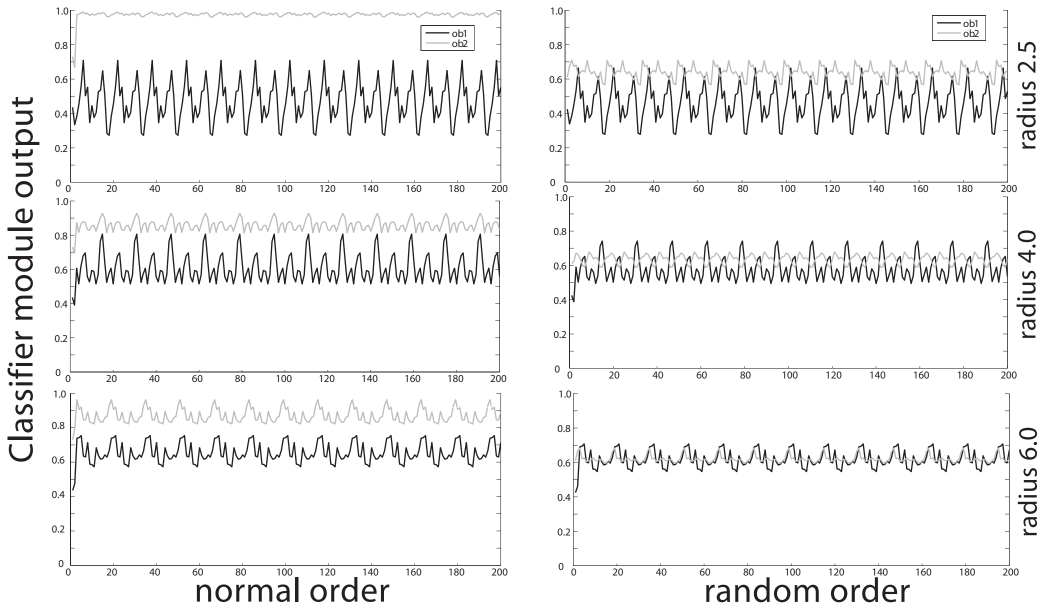

Figure 6.8: RBF model activity for different radii of strategy 3. The left column shows the activity when using ordered training views. The right column shows the activity when the order of the training views was randomised. The first row shows the activity when using the same radius during training and testing. The middle row shows the activity when the radius of the testing strategy was increased to 4. Finally the bottom row shows the activity when the radius is 6.

Figure 6.8 shows that the RBF model exploits the temporal structure when using the same strategy during training and testing even when the radius of the trajectory is double the radius used during training. When the radius is the same during training and testing (top row), the recognition signal drops significantly when the order of the training views is randomised. Even when the model activity shows that the recognition signals become more similar when the radius is increased (middle and bottom rows of the left column), there is a significant difference in signals for object 1 and object 2 compared to the case when the order of the training views was randomised (middle and bottom row of the right column). Therefore, when using strategy 3 during training and the radius is increased during testing using the same strategy, the RBF model can still exploit the temporal structure in the DBCV.

6.5.2 Moving the centre of strategy 3

In this case, the radius of strategy 3 was kept constant during training and testing. However, the centre of the trajectory was moved during the testing phase. Figure 6.9 shows the RBF model activity for different cases where the centre of the testing trajectory is placed away from its position in training.

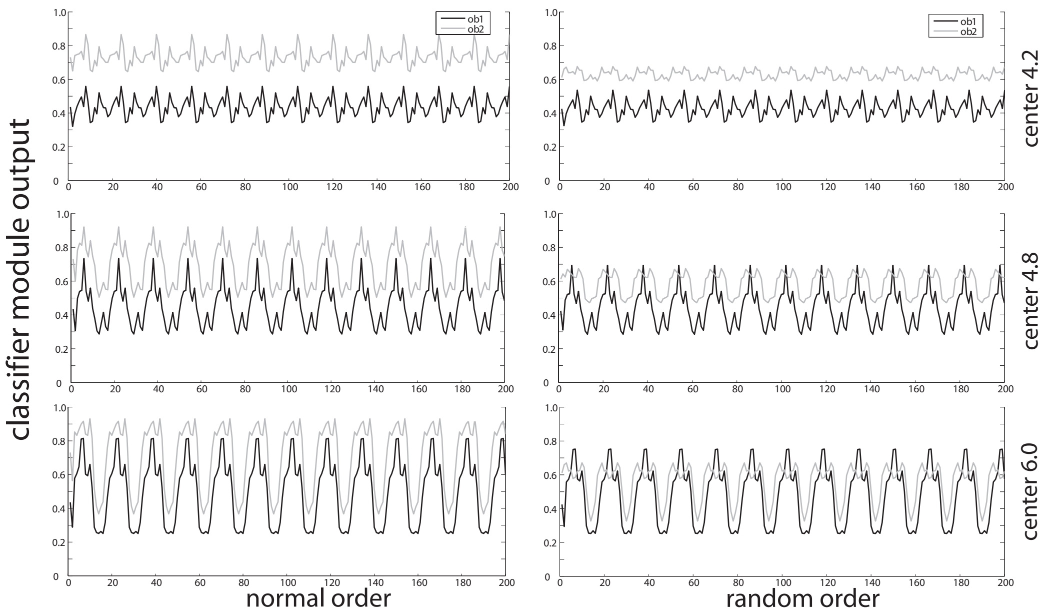

Figure 6.9: RBF model activity when the centre of the strategy 3 was moved during the test phase. The left column shows the activity when using ordered training views. The right column shows the activity of the model when the order of the training views was randomised. The figures in the first row show the activity when placing the centre at (0, 4.2). The figures in the middle row show the activity when the centre was placed at (0, 4.8) and the figures in the bottom row show the activity when the centre was placed at (0, 6).

By changing the centre of the trajectory during the test phase, the distance between the object and the agent was different along the test. Therefore, the scale of the object views changed continuously along the trajectory, as well as the point of view. Since this variation was not present during training, this is reflected in the exploitation of the temporal structure of the DBCV. Even when the figure shows that the RBF model exploits the temporal structure when using the DBCV, the difference between the model activity in normal order versus random order is less evident than in the previous case when the radius was increased during testing.

6.5.3 Using strategy 3 for training and the testing trajectory for testing.

In this case the trajectories during training and testing were different. Figure 6.10 shows the activity of the RBF model when trained using strategy 3 and tested using the testing trajectory. In this test, both objects are present. During the first part of the test, object 1 is within the field of view (period 1) and during the last part of the test, object 2 is within the field of view (period 2). For this test only the training views of object 2 were considered with normal and random order. Therefore, only the activity corresponding to object 2 (grey lines) changes.

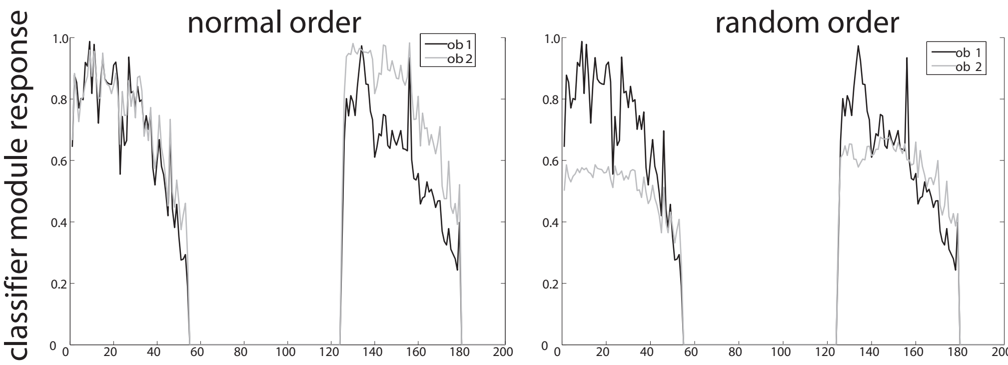

Figure 6.10: RBF model activity for the testing trajectory when using strategy 3 during training. The left column shows the activity when using ordered training views. The right column shows the activity when the order of the training views of object 2 are randomised.

The difference in the model activity between the normal (left column) and random (right column) orders of the training views shows that the RBF model exploits the temporal structure of the DBCV. However, in the previous experiment where the model was trained using strategy 4 and tested using the testing trajectory, the difference in the activity between the cases when using normal versus random order seemed less distinctive (see figure 6.7). In the next section I investigate this by gradually modifying strategy 4 to find out why the exploitation of the temporal structure of the DBCV decreases when using strategy 4 during training and the testing trajectory.

6.5.4 Moving the centre of strategy 4

The first modification made to strategy 4 during testing consisted of moving its centre. The RBF model activity in figure 6.11 shows that in this case the exploitation of the temporal structure of the DBCV decreases significantly when the centre of strategy 4 is moved during testing in comparison with the case when strategy 3 was used.

The activity of the RBF model in this case shows that there is practically no exploitation of the time structure of the DBCV. An important issue to consider in this case is that by moving the centre of strategy 4, the distance between the agent and the object increases and decreases continuously (see distance in the figure 6.9). In contrast, during training, this distance only increases. Therefore, the exploitation of the temporal structure is significantly affected, even when the time intervals during training and testing are

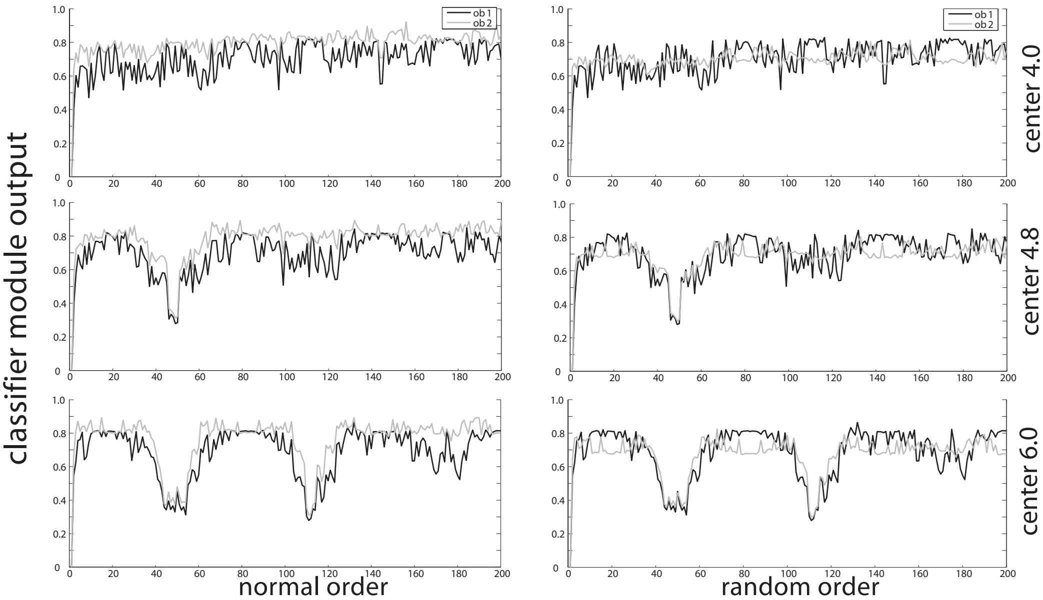

Figure 6.11: RBF model activity when trained and tested using strategy 4 and when the centre of the trajectory is moved during the test. The left column shows the activity when ordered views were used during training. The right column shows the activity when views were randomised during training. The top row shows activity when the center of the testing trajectory was at (0, 4) (as in learning). Middle row shows activity when the centre of the trajectory was at (0, 4.8) and the bottom row shows activity when the centre was at (0, 6.0).

the same. Another issue that changes between training and testing is the time intervals in which the views are taken. During training, the time interval is 10 time steps, while during testing, the time interval is 1. Therefore, the amount of change in consecutive views could be so large that the temporal structure could not be exploited. This issue is investigated in the next case.

6.5.5 Considering interval timing for strategy 4

In order to evaluate the role of time intervals at which the views were taken, I analyse the model activity for two conditions. One when the time intervals are the same during learning and testing, and when the time intervals are 3 time steps during training, and 1 time step during testing.

Figure 6.12 shows the activity of the RBF model using similar intervals during training and testing.

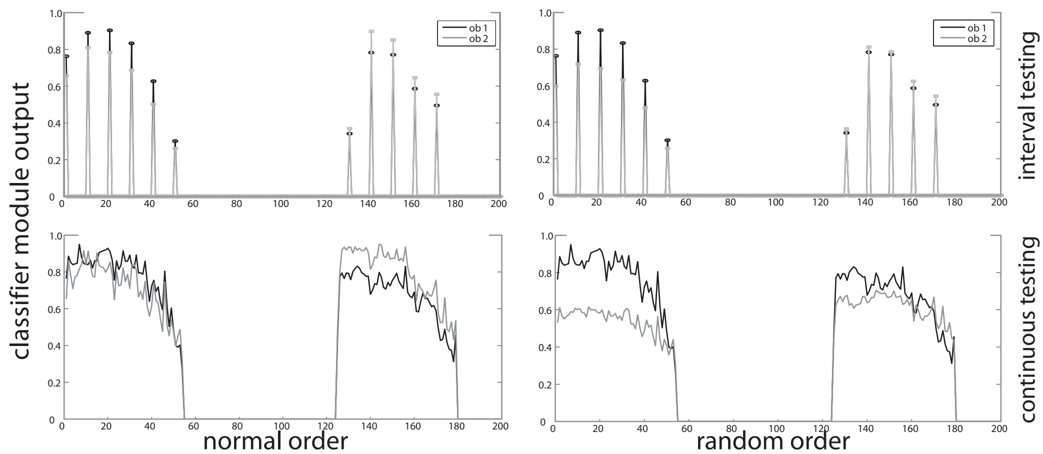

Figure 6.12: RBF model activity during testing trajectory. The model was trained using strategy 4: In the top row, the interval between each view is 10 time steps during training and testing. In the bottom row, the interval was 3 time steps during training and 1 time step during testing (continuous). The left column shows the activity when the model is trained using ordered views and the right column shows the model activiy when using randomised training views.

In the first case (top row in the figure), the interval is 10 time steps during learning and testing phases. In the left column, the RBF model shows the activity when the training views of object 1 and object 2 were ordered. In the right column the training views of object 2 were randomised. The difference between the left and right columns of the first row shows that, when the interval is the same during training and testing, the model does exploit time dependency when using the DBCV. In the second case (bottom row in the figure), the interval is 3 time steps during the training phase and 1 time step during the testing phase. In this case, the difference between the left column and the right column shows that the RBF model does exploit the time dependency when the intervals are similar. Therefore, the exploitation of the temporal structure was significantly affected when strategy 4 was used in the previous cases.

6.6 Experiment 4: Using more objects

In the previous experiments it was demonstrated that the RBF model can exploit temporal structures in the DBCV to perform object recognition. However, the possibility that movement can provide advantageous points of view to aid visual discrimination for the case where similar objects were being analysed was left unexplored.

In certain conditions, two objects can be very similar from a particular point of view. A single view might not contain enough information to allow the visual system to recognise a particular object unambiguously. In the real world, this problem is often resolved by motion which provides visual systems with different perspectives that help to discriminate objects. As a way of mirroring this, active vision in artificial visual systems allows for the purposive control of sensors which provides multiple views. This experiment presents a study of the RBF model where the use of movement strategies represents an advantage to discriminate similar objects.

First, in order to study the temporal information exploitation in the recognition process for tasks where rotation of objects was important for discrimination, a third object was used in the experiment. The third object is very similar to object 2 (bolt), however, in one of its sides it has an extra branch. This object (bolt 2), therefore, needs to be seen from a perspective where the extra branch can be seen so that it is distinguishable from the original bolt. After running the experiment with three objects, I consider three additional different objects in the object recognition tasks. In this case I compare the activity of the model when using the DBCV when the training and testing trajectories were similar versus the case when they were different.

Using 3 objects. In figure 6.13 an example of a training view for each object is shown. Object 3 is very similar to object 2. Object 4 is a simple model of a house. This object has significantly more texture than the others, with bricks and rock textures in the walls, a wooden door, a roof texture and windows. Object 5 is an extinguisher, from some views this object is similar to objects 2 and 3. Object 6 is a model of a webcam.

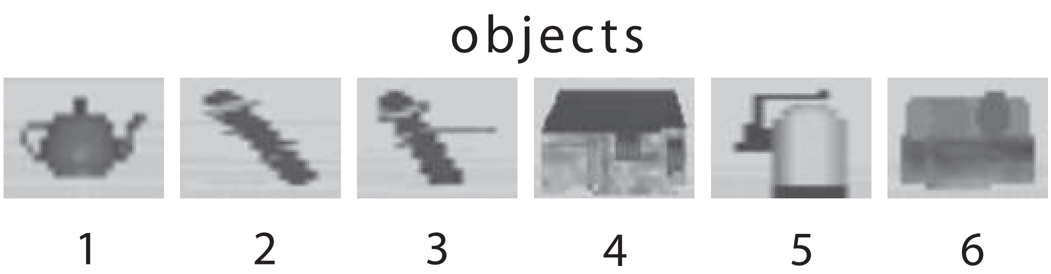

Figure 6.13: Examples of views of the objects. Object 1: teapot, object 2: bolt1, object 3: bolt2, object 4: textured house, object 5: extinguisher, object 6: webcam.

At first, objects 1,2 and 3 were used to train the model and only objects 2 and 3 were used during the test. In order to discriminate between objects 2 and 3, it is necessary to move so that the difference between these two objects becomes significant.

When the way the objects change along the trajectories is not very similar during training and testing (left column), the model activity shows that the RBF model using

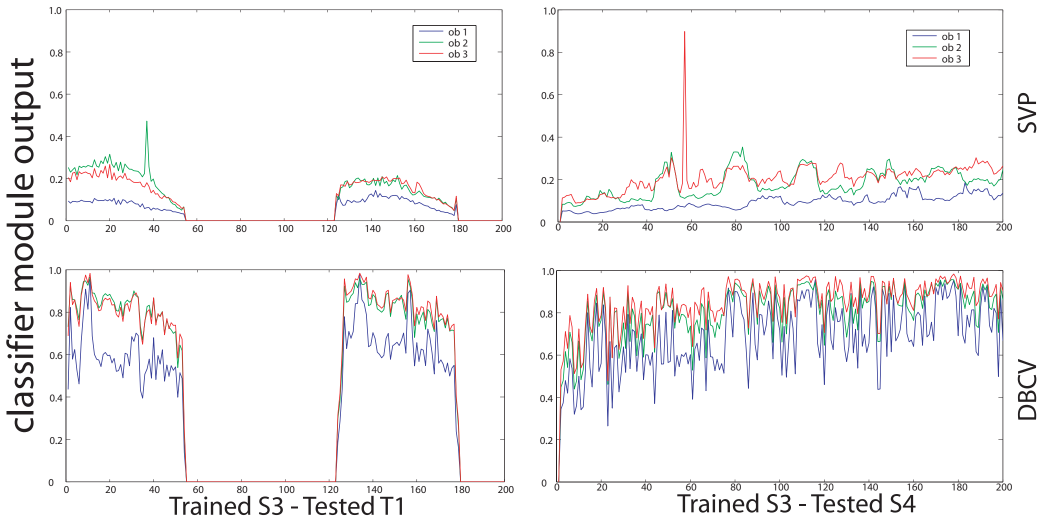

Figure 6.14: Model response during the testing phase. The first row shows the activity of the RBF model when a single view presentation (SVP) was used. The second row shows the model activity when DBCV was used. The left column shows the model activity when the model was trained using strategy 3 and tested using the testing trajectory. The right column shows the model activity when the model was trained using strategy 3 and tested using strategy 4.

DBCV is not distinctively better than when using SVP (the response for object 3 is not very different than the response for object 2 in both cases). However, when the way the objects change along the trajectories is similar during training and testing (right column), using the DBCV is significantly better than using SVP.

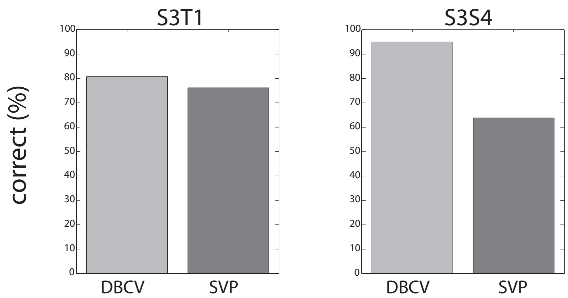

Figure 6.15: Comparison of the performance of the RBF model for DBCV and SVP conditions. The percentage corresponds to the average number of the total correct classifications during the test phase. For the trajectory 1 (left graph) the total number of presentations is 110 (55 in period 1 and 55 in period 2). For strategy S4, the total number of presentations is 200. Since the model is deterministic, the bars only represent the average of the correct guesses over the total number of views (presentations) during the test trial.

Figure 6.15 shows a comparison of total number of correct guesses by the RBF model when using SVP and DBCV. This figure shows that the performance of the RBF model using DBCV can improve the recognition process when the views of the objects change along the trajectories in a similar way during training and testing. The percentage corresponds to the average number of correct object recognition guesses during the test phase. For test trajectory T1, the number of presentations is 110 (55 during period 1 and 55 during period 2). For movement strategy S4, the number of presentations is 200.

In order to test whether the RBF was using the temporal structure of the DBCV when using six objects, the order of the views was randomised during training. The results show than even when using more objects, the RBF model does exploit the temporal information when using DBCV and can correctly discriminate between objects that are very similar from multiple points of view (objects 2 and 3. See figure 6.13) only if the trajectories during and training and testing are very similar.

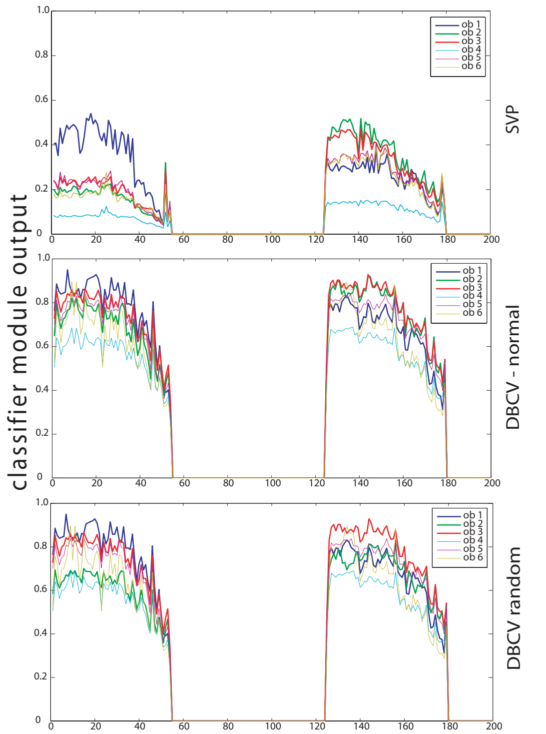

Figure 6.16: RBF model activity during the testing phase and using strategy 4 during training. Object 1 was present in period 1 and object 2 was present in period 2. This time the model was trained using 6 objects.

Figure 6.16 shows the RBF model activity when strategy 4 was used during training and the testing trajectory was used during the testing phase. In this case, when the trajectories are different during training and testing, the model fails to correctly discriminate between object 2 and 3 when using the DBCV. This is in contrast to the case when using SVP (top row). However, the difference between the model activity when the order of the views was random (bottom row) and normal (middle row) shows that the RBF model exploits the temporal structure of the DBCV. Having different movement strategies during training and testing affects the RBF model when using DBCV because the way the views change is different between training and testing, which is what the DBCV measures.

In contrast with the previous case, the RBF model activity was also analysed when the training and testing trajectories were similar and six objects were used. In this case, object 3 was present during the test phase. Figure 6.17 shows a comparison of the RBF model activity for the case where the movement strategies are similar during training (strategy 3) and testing (strategy 4) when using SVP and DBCV.

In this case, the model is trained using strategy 3 and tested using strategy 4 (these two movement strategies are similar). The activity in figure 6.17 shows that in this case, the RBF model can correctly discriminate between object 2 and object 3 when using

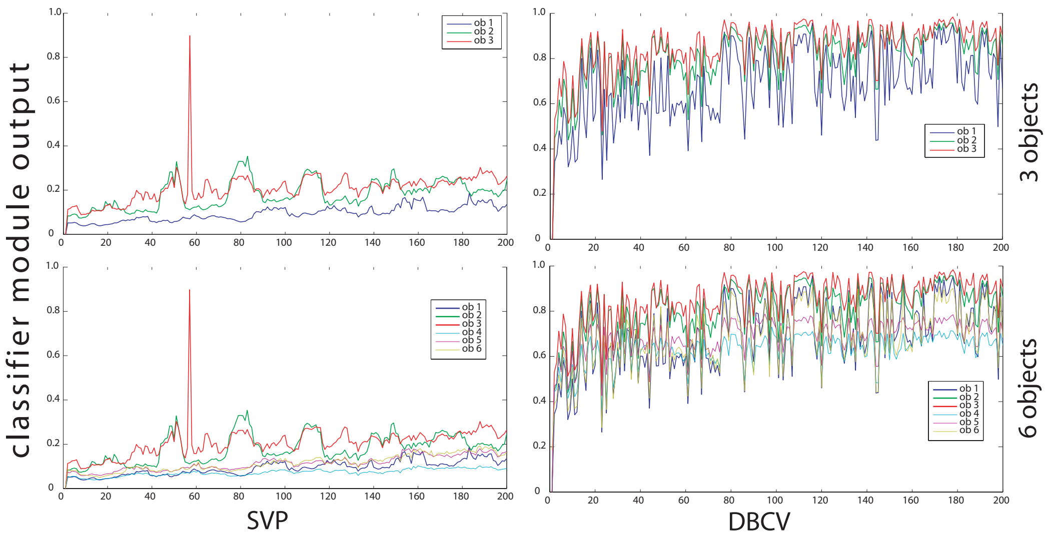

Figure 6.17: RBF model activity comparison between the cases when using 3 or 6 objects during training when using SVP and DBCV. For this experiment, object 3 was present in the arena during the test phase.

DBCV. In contrast, when using SVP, the model fails to correctly discriminate between these objects along the trajectory during testing. In this case, the exploitation of the temporal information proves to be useful in the discrimination of similar objects.

6.7 Discussion

In this chapter four experiments were carried out. In the first experiment it was demonstrated that there are differences in the efficacy of movement strategies for acquiring visual information with the RBF and HMAX models. It was demonstrated that when the training views provide enough variation in rotation and scale, the RBF model outperforms the HMAX model. The reduction in the recognition performance of the HMAX model when using strategies that provide significant variation in rotation and scale, shows that the reduction of the dimensionality and the generalisation by the HMAX (explained in the previous chapter) decrease its capability to discriminate between two simple objects when a BDM is used (see similarity maps in the first experiment). In contrast, the increase in the performance of the RBF model when enough variation in rotation and scale are provided through the movement strategies during the training shows that this model increases its discrimination capabilities when using an active BDM due to its high specificity.

In the second experiment the difference between consecutive views was used to represent temporal structure in the presentation of visual information. In this experiment it was demonstrated that the RBF model can use the DBCV to perform object recognition. It was also shown that the degree of exploitation of the temporal structure in the DBCV changes with the employment of different movement strategies. In some situations the RBF model using the DBCV performed object recognition correctly even when the temporal structure in the DBCV was not exploited by the RBF model. When only two objects are used, the DBCV was smaller for views of the same object than between views of object 1 and object 2. The model fails to correctly discriminate the objects when this is not the case. Potential problems can arise when using more objects where the DBCV for the same object is not smaller than views for different objects. In these cases, the use of the SVP shows a better recognition performance than when using DBCV.

In the third experiment, the robustness of the RBF model was tested to perturbations in the trajectories when using DBCV. This experiment shows that the robustness of the RBF model to perturbations in the trajectories depends on the nature and amount of change that the views of the objects go through during training and testing. These results further explain the conditions that determine the similarities in these changes. For example the amount of change can be determined not only by movement itself but also by the intervals in which the views are taken during training and testing. Additionally, studying the robustness of the models to perturbations in the testing trajectories is important because it can shed some light on the characteristics of the required movements of the agent during testing. For instance, if the model is robust to perturbations to the testing trajectories for a particular task, then the motor control of the agent during testing would not need to be so restrictive. In that case, it could be easier to find a controller that could solve such a task. In contrast, if the recognition process is severely affected by perturbations of the testing trajectories, the required movements would need to be particularly precise and would impose additional restrictions on the controller. These experiments show that when the amount of change in consecutive views is significantly different during training and testing, the RBF model shows a better recognition performance using SVP than when using DBCV.

In the fourth experiment, multiple objects were considered. In particular it was shown that for objects that are very similar from most points of view, the exploitation of the temporal structure in the DBCV can be advantageous for the RBF model when the trajectories during training and testing are similar. This experiment also demonstrates that the exploitation of temporal structure by RBF model can be used when more objects are considered. In this case, recognising the way the objects change between consecutive views can be advantageous over a single view presentation, particularly for objects that look very similar from multiple points of view.

6.8 Conclusion

In this chapter it was demonstrated that there are movement strategies that can provide enough variation in the visual information during training so that the RBF model can outperform the HMAX model. It was also demonstrated that the DBCV can be exploited to improve the recognition performance of the RBF model when the training and testing trajectories are similar. Additionally, it was demonstrated that the robustness of the RBF model depends on the similarity between the changes that consecutive views undergo during training and testing. This similarity is not only dependent upon the way the agent moves during training and testing, but also upon the time intervals when the consecutive views are taken. Finally, it was demonstrated that the DBCV can be used by the RBF model to perform object recognition when multiple points of view are required to discriminate similar objects and also when more objects are used.

The experiments in this chapter provide evidence that the RBF model is a good candidate for autonomous mobile agents performing object recognition. This is because when enough variation is provided during training through the movement strategies, the RBF model can perform object recognition in a mobile agent reliably. Additionally, the use of temporal information by the RBF can be beneficial when movement strategies are similar during training and testing, even with multiple objects, and also because the model is robust to perturbations in the trajectories during testing.

Chapter 7

A simple model of object recognition in the real world

7.1 Introduction

In previous chapters it was demonstrated that in simulated conditions the RBF model could perform active object recognition reliably when aided by an attentional mechanism. Specifically, the RBF model exploited the variation in scale and rotation in the visual information acquired by four different movement strategies. It was also demonstrated that this model is robust to perturbations in the visual system as well as to perturbations in the trajectories used to acquire the visual information. These results suggest that this model is a good candidate to perform object recognition in an autonomous mobile agent, at least in simulation. However, to prove its worth, research work of this type must have its predictions validated in the real world.

In the field of robotics research, there has been much discussion about the advantages and disadvantages of carrying out experiments in simulation as opposed to using real robots. On the one hand, simulations allow us to cross the "real time gap" where processes are not restricted to function in real time but can work on a faster time scale. This aspect can be a great advantage in Evolutionary Robotics (as was shown in previous chapters). Simulations can also provide the advantage of carrying out experiments in extremely well controlled conditions, and reduce the costs of building and maintenance. On the other hand, it has been argued that when using simulations there is the risk of studying problems that actually do not exist in the real world. This can be a serious problem when the research being conducted is meant to help to understand phenomena in the real world. There is also a concern that by using simulations, there is a chance that the solutions or explanations do not translate into the real world (Brooks, 1992; Floreano et al., 1998).

In order to validate the results of the previous chapters, in this chapter I therefore implement experiments in real world conditions. However, these experiments differ from the ones carried out previously in simulation as the real world is much more complex for visually guided agents. This is primarily because the real world is a very noisy environment compared to a simulated experiment. First of all, the visual information in the real world is very noisy. For example, there are shadows, different illumination conditions, reflections, etc., that were not present in the previous simulated experiments. Secondly, the motors and sensors are also noisy, making it difficult to control the interaction between visual sensors and motors with arbitrary precision. Therefore, when working with real world biomimetic visually guided robots, there are certain aspects that need to be restricted to be able to study the particular problem we are interested in. The main restriction here is the use of a gantry robot (to be described in detail below) rather than a simulated wheeled robot. This allows us precise and therefore repeatable control of movement. This allows us to focus on the impact and interaction of environmental and sensor noise. While motor noise also has an impact in the real world, its nature will depend on the type of robot used and focus on this aspect is not relevant to this thesis.

In this chapter the RBF model is evaluated in three real world object recognition experiments, using seven different objects. In the first experiment the model is tested using single view presentation (SVP). In the second experiment the exploitation of temporal information by the RBF model was tested using the difference between consecutive views (DBCV). In both experiments the exploitation of the variation of the visual information in the object views is analysed using four different movement strategies. The results of these experiments validate the predictions made in the previous chapters. The object recognition performance of the RBF model is influenced by the variation in the object views acquired during movement. Furthermore, this movement-defined variation determines regions in the arena where the recognition performance of the model improves.

7.2 Methods

7.2.1 Experimental set-up using the Gantry robot

The experiments in this chapter were carried out using the Gantry robot: a large precision Cartesian robot with a panoramic camera in a 3D arena measuring 300cm $\times$ 200cm $\times$ 200cm. Seven objects were placed individually in the center of the arena and images were taken from each vertex of a grid of 290/5 cm $\times$ 170/5 cm positions, except for a circular buffer region around the centre of the arena (where the object was placed) measuring 25 cm (see inset in figure 7.1).

As with the previous experiments carried out in simulation, the experiments in this chapter consisted of two phases. During the first phase the RBF model was trained using views collected by the Gantry robot from grid positions specified by one of four movement strategies. This phase was performed for each of the seven objects individually, resulting in seven trained networks. During the testing phase the model was presented with object views collected from every location in the arena. At each location, the seven trained networks are presented with a view of all objects, one at a time. The network with the highest output indicates which trained object the presented object is recognised as.

The experiments use the toy-like objects shown in figure 7.1: a black truck, potato man, ninja, truck, squirrel, frog, and red truck. The set of images for each object formed an object image database with 2004 images, giving 14, 028 images in total. In figure 7.1, the different objects used in these experiments are shown. The characteristics of the objects

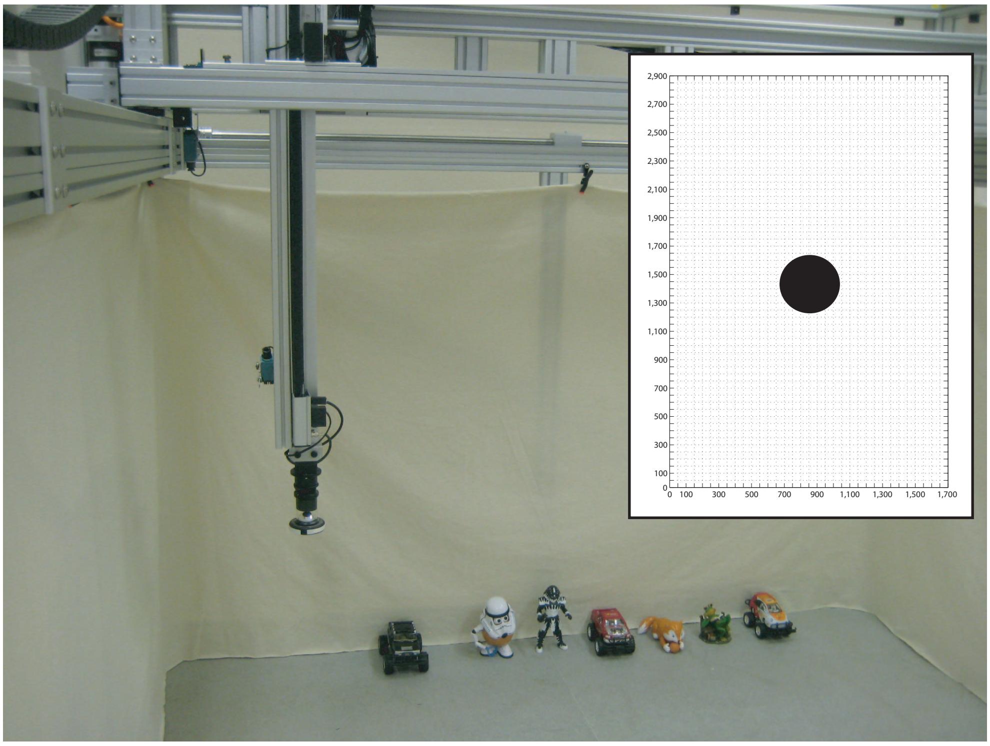

Figure 7.1: Gantry robot and the seven objects used for the object recognition tasks. At the end of the mechanic arm, there is a panoramic camera. Inset: 1700 × 2900 mm arena. The black circle represents the circular buffer where the object was placed.

are intentionally chosen so that the recognition tasks were not trivial. For example, there are similar objects in one category (three trucks with similar shapes). Also, the colour intensities of the different objects are similar in several cases (the potato man and the squirrel).

Although the visual conditions in the arena are controlled to some extent, a plain background (not uniform though), nearly constant illumination conditions (not uniform though) and non-cluttered scenes, this is not a trivial task. It is important to note that the purpose of this chapter is to evaluate whether the intuitions from simulation carry over to reality. A further step in the investigation of this model in real world conditions would be to study the model in more challenging conditions.

7.2.2 The visual system

The visual system of the RBF model used in this chapter is the same as in the simulated cases, namely the visual system consisted of two modules, an analysis module and a classifier module (see sections 3.2.1 and 3.2.2 for a detailed description). However, the visual information in this case is acquired by a real panoramic video camera. The video stream from the panoramic camera consisted of a series of 360 degrees panoramic colour images.

Visual information The panoramic images were converted into grey scale images. These images were "unwrapped" (the algorithm is described in table 7.1) and cropped into 360 × 400 pixels images. Figure 7.2A shows an example of a panoramic image taken from one particular position using the 'redtruck object'. In figure 7.2B the unwrapped version of the image is presented.

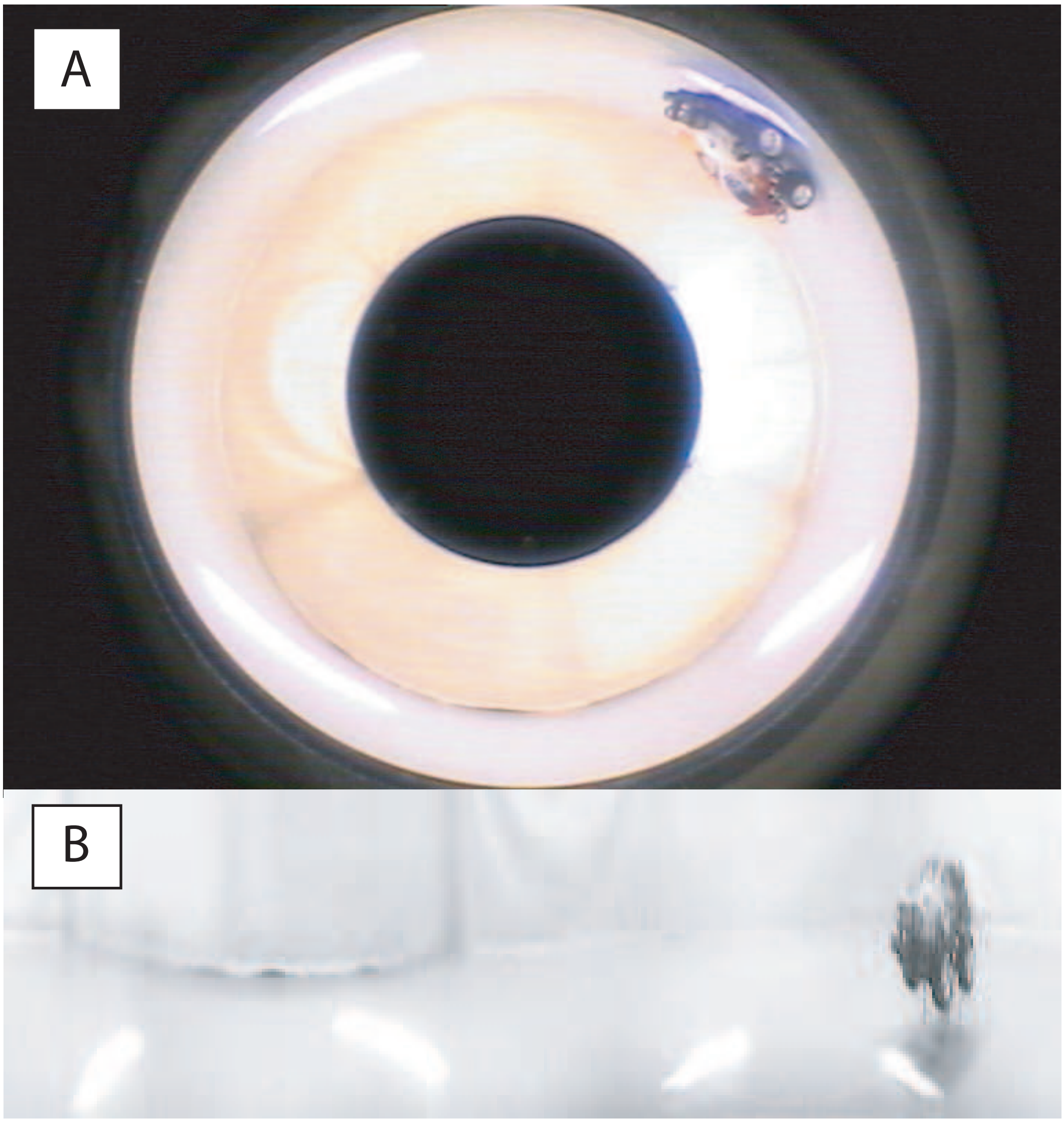

Figure 7.2: Example of a panoramic image and the corresponding unwrapped and cropped image.

Blob detection mechanism. A blob detection mechanism (BDM) was implemented to select salient features in the images. This mechanism was the same as the one used in section 3.2.3 (and in simulations in chapters 5 and 6). Very briefly, this blob detection mechanism consisted of selecting salient regions by dilating fragments found by an edge detector (see section 3.2.3 for details). In order to select a particular detected blob in the visual field, three "attention criteria" were used: 1) the order in which the blobs detected. The last blob detected was selected since most of the time the last blob corresponded to an object. 2) the pixel intensity of the blobs detected. The least brightest blob detected was selected since most of the artifacts were reflections which had a very high pixel intensity. 3) The last criteria was the eccentricity of the detected blob. In this case the most rounded blob was selected since the shape of most of the reflective artifacts were thin and large. The best performance of the BDM was obtained using the third criteria.

1) Select the centre of the panoramic image (which is determined by the camera settings).
2) Set r in [100, 400] and theta in [1, 360] (which corresponds to the region of interest in the panoramic images (figure 7.2).
3) For each r and theta, use the centre of the image to:
a) calculate where in the array of pixels r, theta is.
b) based on the existent data and the position you want, calculate the gray-scale value through a 2-dimensional bicubic interpolation (using matlab function `interp2`).

Table 7.1: Unwrapping algorithm to convert panoramic images into landscape images.

Accuracy of the BDM Five different artefacts triggered the detection of blobs in the visual field, namely: an object; parts of an object; a reflection or set of reflections; a shadow; parts of the arena. Only the first is a correct blob identification. Once the blobs were selected, they were cut and normalized to a size of 80 × 60 pixels. In contrast to the simulated case, this blob detection mechanism was quite noisy, only 60-80 % of the times the correct blob was selected (for the best attention criteria). In the other cases, shadows, reflections or part of objects were selected as blobs and passed to the analysis module.

In order to measure the accuracy of the BDM used in these experiments, 100 random detected blobs were manually evaluated to see if the object was correctly identified by the BDM. This process was carried out five times. The average accuracy for the BDM when using the eccentricity criteria was 80% with a standard deviation of 4.47.

7.2.3 Movement strategies

In the previous experiments, the trajectories of the agents were hard-wired. That is, there was no explicit interaction between sensory information and motor control, and images processed during the simulation were taken every 10 time steps. An image database was built for every possible position of the robot in the arena by taking images from a grid of points (see the inset in figure 7.1). The object view to be processed was determined by one of four movement strategies (as shown in figure 7.3).

For strategy 1, 16 points (in a vertical line) in the arena are used to select the 16 training views from the databases. For strategy 2, 16 points (in a horizontal line) in the arena, are used to select the 16 training views from the databases. For strategy 3, 16 views around the object were collected following a circular trajectory and finally, for strategy 4, 16 views were collected around the object following a spiral trajectory.

The left column of figure 7.3 shows the different movement strategies used to collect the training views in the following experiments. Note that because of dimensions and the discretisation of the arena into a grid of points, the strategies are slightly different to the ones used in the simulated experiments. For example, strategy 1 does not go straight to the object but, slightly to its left. Therefore, strategy 1 provides not only scale variation but also rotation variation in the object views. This is in contrast to the simulated case where this strategy only provided variation in scale. In strategy 2, the scale variation is significantly larger than in the simulated case. Strategy 3 is positioned very close to the object in comparison with strategy 3 in simulation. Finally, strategy 4 in this case has a scale variation significantly smaller than in the simulated case.

During the testing phase, in some of the following experiments the positions in which the test views are collected are restricted to a particular region in the arena. This region, called "the advantageous zone" is a ring around the object (see figures 7.4). The shape and dimensions of the region were chosen arbitrarily based on the general performance of the RBF model where most of the recognition signal was correct (that is, the positions where the object in the arena was correctly recognised). This particular set up for the advantageous zone was selected based only on the performance of the RBF model when using SVP, although it was also employed to evaluate the model using the DBCV (see figure 7.6). Optimisations of the parameters of the region could be based on the levels of the recognition signals, for example using a threshold for the recognition signals to determine its limits.

7.3 Results

Two experiments were carried out in this chapter with the primary objective of validating whether the RBF model can operate in the real world. Specifically, the first experiment investigates how the variation in the training views due to different movement strategies can be exploited and whether performance varies in different parts of the arena. The second experiment performs a similar analysis to investigate whether temporal information can be exploited by the RBF model in the real world.

7.3.1 Experiment 1: Movement strategies and RBF model in the real world

This experiment consisted of training the model using single view presentation (SVP). The training views are collected using the four different movement strategies. The model was then tested in two conditions, first in every possible position in the arena and secondly, in a restricted region in the arena (the "advantageous zone" previously described). The results of the first case are shown below in a recognition map (figure 7.3). A recognition map is a grid with every possible position in the arena. For each training trajectory, the model is tested by presenting views of all seven objects for every position in the arena. The warmer (redder) the colour of a particular position, the more correct guesses the model made in that position.

The recognition maps in figure 7.3 show that the RBF exploits the variation in the visual information provided by the four different strategies. The distribution of the recognition performance in the figure shows that there is a zone in the arena where most of the strategies are better exploited.

In this experiment the recognition maps show that the RBF model exploits the variation in scale and rotation in the training views provided by the four movement strategies. This variation in scale and rotation in the training views shaped the regions in the arena where the model has a better performance. For strategy 1, this region corresponds to the area where the training views were collected. For strategy 2, the region where the model has a better performance is more spread out than in the previous case, and this time it is

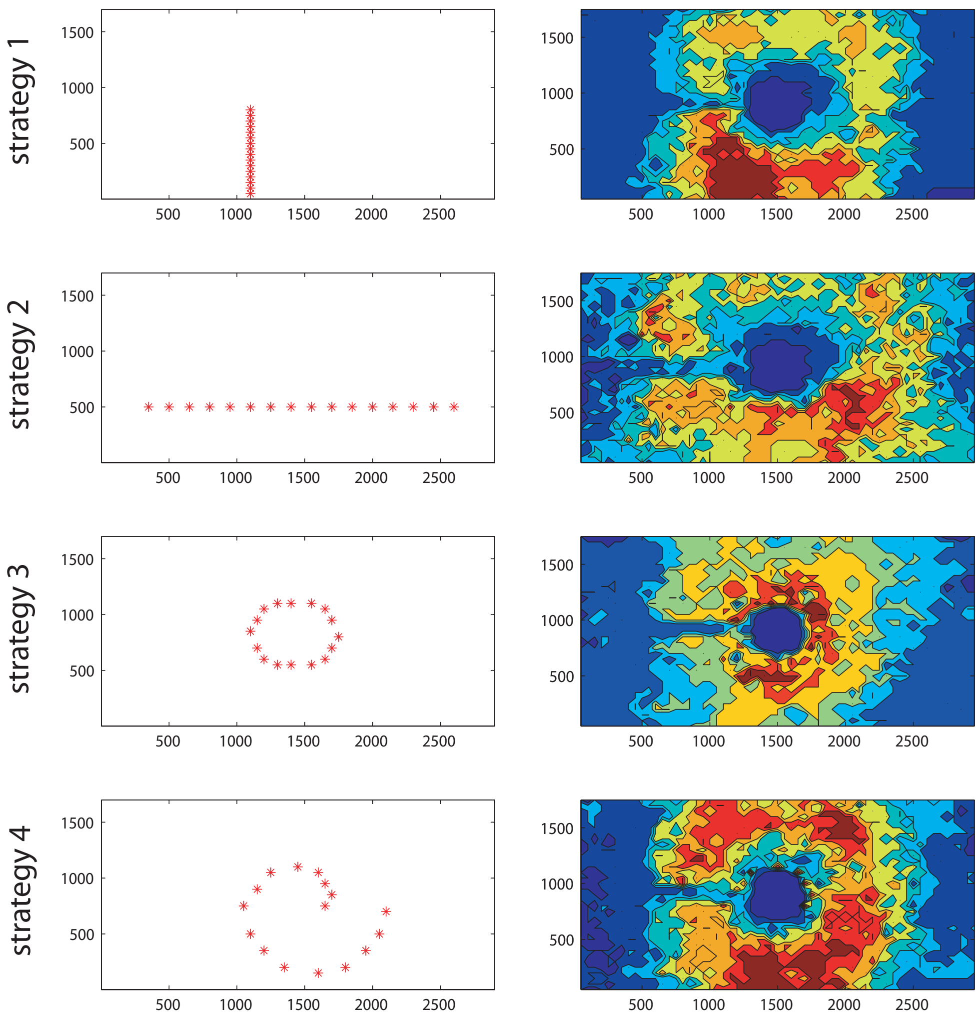

Figure 7.3: Recognition maps for every strategy using the RBF with SVP. The left column shows the four movement strategies to collect the training views.

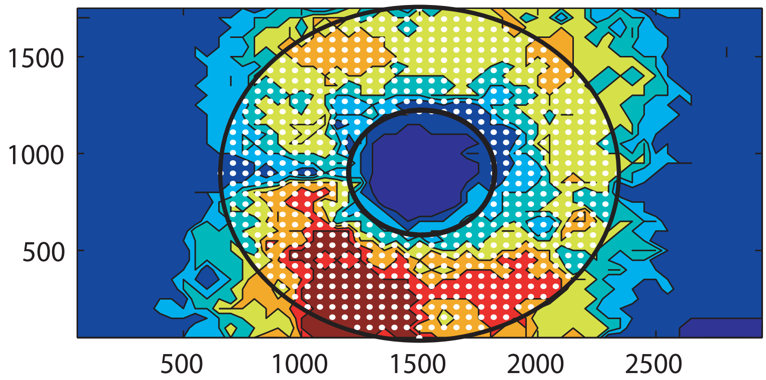

Figure 7.4: Advantageous zone. Recognition map when using SVP and strategy 1.

more horizontal rather than vertical, which again reflects the way the training views were collected. This is due to the fact that in strategy 2, the variation in scale and rotation was significantly larger than in strategy 1. In general, the maps for strategy 1 and strategy 2 show significant red regions only from the side of the arena were the training views were collected. That is, the performance of the model is poor when trained using these strategies because the recognition of the objects when evaluated from points of view which were not present during training becomes ambiguous.

For strategy 3, the region where the model has a better performance is very similar to the area where the training views were collected. In this case, the scale variation was nonexistent, but the rotation variation was larger than in the previous cases. However, we see that the small variation in scale translates into a less spread out region where the performance of the model is better. For strategy 4, where the scale and rotation variations were relatively high, the region where the performance of the model is high, is more uniform and spread out. The increase in variation in rotation (strategies 3 and 4) is reflected in the improvement in the recognition performance from multiple directions (surrounding the object in the map), in contrast with the case when the rotation variation is low (strategies 1 and 2). In the case where variation in rotation is low, the performance is better only in the side of the arena where the model was trained. However, the region where the performance of the model is increased, is expanded and spread out when a significant scale variation is present during training (strategies 2 and 4).

The second condition for this experiment consisted of evaluating the model only in the advantageous zone. The restriction of the region in the arena where the model is evaluated followed the premise that a simple movement strategy could be used so that the agent approaches this sub-region of the arena (for example, object approaching using optic flow (Duchon et al., 1998; Young, 2000)). The advantageous zone was selected based on the regions in the recognition maps where 'roughly' the model shows a better performance for most of the strategies. When the model was tested in this case, only the points within this region were considered (see white points in figure 7.4).

Figure 7.4 shows the advantageous zone for the recognition maps when using SVP. In the figure, the advantageous zone overlaps the recognition map when using strategy 1 during training.

Strategy	general (%)	adv (%)
1	41.70	61.60
2	50.68	47.95
3	37.92	56.62
4	50.82	74.88

Table 7.2: RBF model performance. The left column shows the performance when it was evaluated in every position in the arena. The right column shows the performance when the model was evaluated only within the advantageous zone.

The performance of the RBF model for every movement strategy, when evaluated in every position in the arena (left column) and when evaluated only in the advantageous zone (right column), is shown in table 7.2. An increase in the model performance shows that being in that zone during the testing phase represents an advantage for the recognition process except for strategy 2 where the performance drops slightly when the model is evaluated only in the advantageous zone. This decrease in the performance of the model in the advantageous zone is due to the large scale variation provided in strategy 2, in comparison with the rest of the strategies (see figure 7.3).

7.3.2 Experiment 2: Using temporal information with the RBF model in the real world

In order to evaluate whether or not the temporal information could be exploited by the RBF model in a real world situation, the difference between consecutive views (DBCV) was considered in this case. Very briefly, the DBCV consisted of taking the absolute value of the difference between consecutive views v_i, v_j after being processed by the RBF model, that is, DBCV$(v_i, v_j) = 1/2 \cdot |RBF(v_i) - RBF(v_j)|$ (as previously explained in section 6.4). In this experiment the training views are also collected using the four different movement strategies. Similarly to the previous experiment, the model was then tested in two conditions, first in every possible position in the arena and second, in a restricted region in the arena. First, we will explain how consecutive views were obtained.

As temporal information is analysed, consecutive views are needed for each position, with the predecessor point determined by a movement strategy. In the training case, the consecutive views were determined by the order in which every position in the movement strategy was worked out: for strategy 1, the first position is the one furthest from the object (see figure 7.5) and the next ones are consecutive towards the object. For strategy 2, the initial view was the furthest left in the arena. For strategy 3, the first view was the one furthest right and the next ones are consecutive views in anti-clockwise direction. For strategy 4, the first view is the one closest to the object and the consecutive views are worked out anti-clockwise. For the last DBCV view, the 16th view and the 1st view were considered, that is, the next view for the 16th view was the first view (as if the sequence of 16 views were cyclical).

The calculation of consecutive views of the test views is performed as follows. Every valid point x in the arena was evaluated by calculating its previous predecessor point y such that, x is the consecutive point of y when the agent was traversing strategy 3 using an interval of 360/16 degrees. When the calculated position y is not valid, either because y is outside the boundaries of the arena or outside the advantageous zone, x was considered invalid. The predecessor point y was calculated in a similar way using strategy 4. However, the results are qualitatively similar to the ones obtained when using strategy 3 and so are not shown.

In the previous simulated experiment in chapter 6, the exploitation of temporal information was possible when the variation in the object views was similar during training and testing. This exploitation of the temporal information was increased when this variation in scale and rotation was significant. The results presented in chapter 6 show that strategies 3 and 4 provide such variation. Therefore, in this chapter only strategies 3 and 4 are used to calculate consecutive views during testing. However, results are also presented for strategies 1 and 2 tested with consecutive views calculated by strategies 3 and 4, as a control to see if it is indeed better to move in the same way during training and testing.

In figure 7.5 the recognition map for the RBF model when using DBCV is presented. Similarly to the previous experiment, this map shows the distribution of correct guesses by the RBF model within the arena. The redder (warmer) the regions, the more correct guesses the model has made in that point. For the points where no colour appears (dark blue), it was not possible to find a predecessor (invalid points).

This recognition map shows that it is better to move in a similar way during training and testing in order to increase the performance of the recognition task. The warmer (redder) regions appear in strategy 3 and strategy 4. However, the model seems to better exploit the temporal information when it is trained using strategy 4 than strategy 3, even though the latter is used to calculate the consecutive views during the test. Given that the consecutive views during testing were calculated using strategy 3, we might expect that the performance of the model would be better when the model was trained using strategy 3 than strategy 4. However, by evaluating the model in every position in the arena, the distance between the agent and the object changes significantly and therefore, a significant degree of scale invariance is also required, favouring strategy 4 which provides both (see last row in figure 7.5). Therefore, the exploitation of the temporal information is determined by the similarity of the changes in the visual information between training and testing. That is, in this case the variation in the visual information was present in scale and rotation during testing, therefore, the strategy that provided such type of variation was only strategy 4.. This is confirmed by table 7.3 which gives the average number of times the model correctly recognises the objects in every valid position in the arena. Strategy 4 outperforms all other strategies, particularly strategy 3.

The recognition map in figure 7.5 also shows that the performance of the model is differently distributed in the arena depending on the way the visual information is presented to the model during training. Therefore, the model was re-evaluated using only positions from the advantageous zone described in the previous experiment (see figure 7.6).

Figure 7.6 shows the advantageous zone overlapping the recognition map when the model was tested using DBCV and strategy 4 during training. The right column in table 7.3 shows the performance of the model using the DBCV when the evaluation was re-

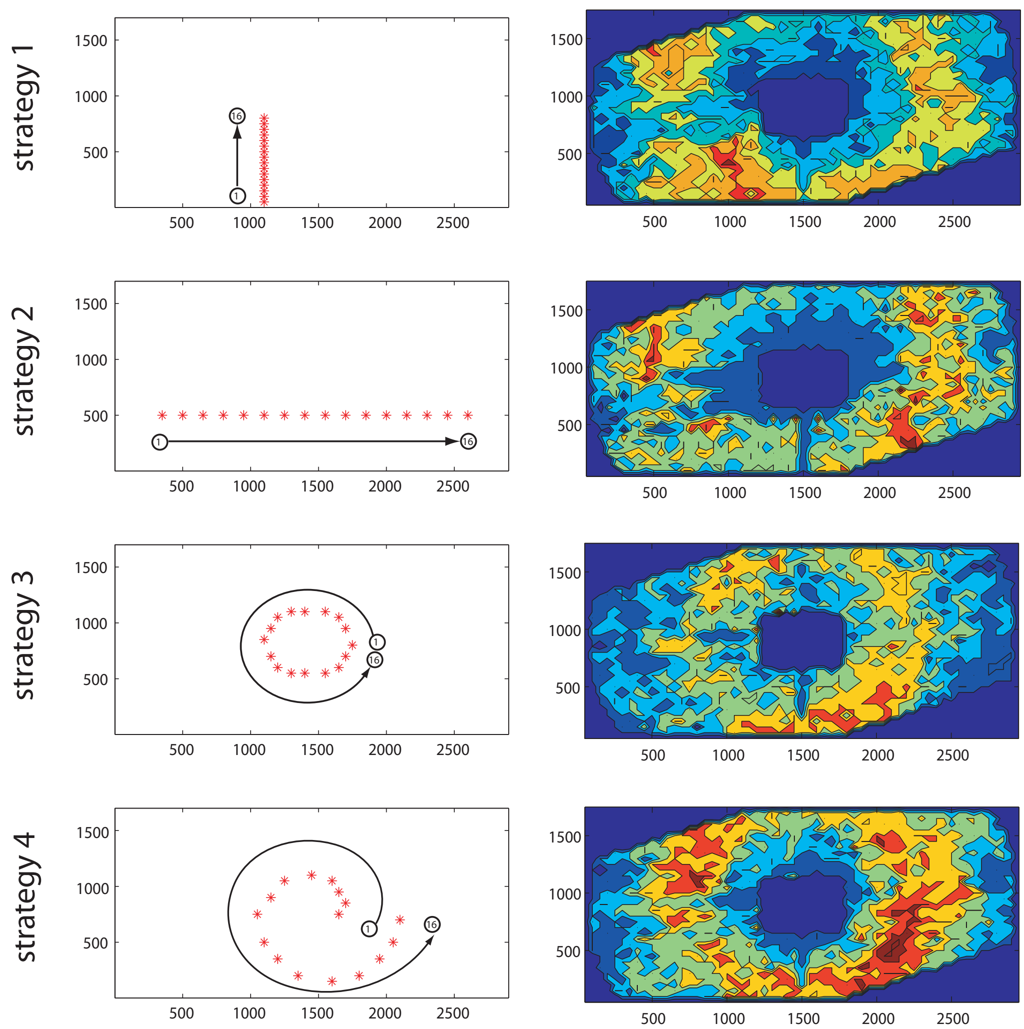

Figure 7.5: Recognition map using temporal information. The circles and solid lines figure shows the order in which the training views were considered when calculating the DBCV. The right column in the figure shows the recognition maps when the model was trained using each strategy and tested in every valid position of the arena.

Strategy	general (%)	adv (%)
1	48.27	43.95
2	42.43	35.48
3	41.28	48.70
4	48.15	49.60

Table 7.3: Performance of the RBF model when using the DBCV. The left column (general) shows the performance in every valid position in the arena and the right column (adv) shows the performance within the advantageous zone.

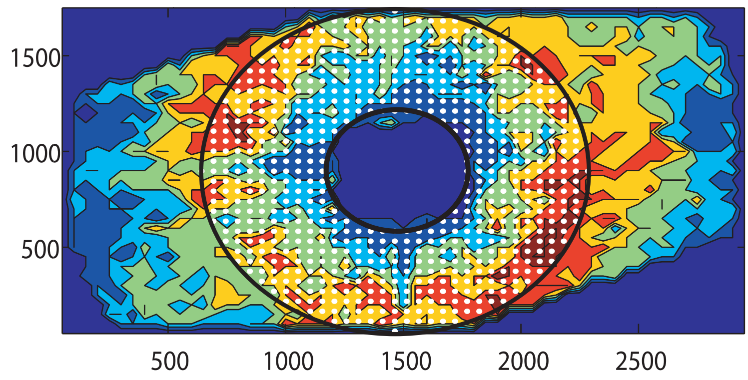

Figure 7.6: Advantageous zone for the recognition maps using DBCV. Recognition map when using DBCV and strategy 4 during training.

stricted only to the advantageous zone. This table shows that when the model is evaluated only in the advantageous zone, the model shows only a small increase in its performance when the variation of the object views is similar during training and testing (third and fourth rows). In contrast, when the model is trained using strategies that do not provide this type of variation during training (first and second rows), the performance of the model decreases. This increase in the performance of the model for strategies 3 and 4 in the advantageous zone was due the fact that in this case there is less scale invariance required since the distance between the agent and the object does not vary as much as in the case where the model is evaluated in every position in the arena. Therefore, strategies that provide significant variation in rotation increase the performance of the model in this case.

7.4 Discussion

The results in the first experiment using SVP show that the RBF model can function in the real world. Moreover, performance of the model increases significantly if the visual information is only considered in the advantageous zone. These results validate the predictions from the simulated case where high variations in scale and rotation are exploited by the RBF model to improve its recognition performance. Thus even when there is a significant level of noise in the BDM, the RBF model performs object recognition in the real world when using SVP.

In the second experiment, the exploitation of temporal information by the RBF model was tested under two conditions. In the first, views were collected in any possible point in the arena during testing. In the second, views were only collected within the advantageous region. Two types of variation in the views were used to calculate the predecessor view in the testing phase when considering the DBCV. In the first one only variation in rotation was used (strategy 3). In the second, variation in rotation and scale were used (strategy 4). However, for both types of variations, the RBF model showed only a slight increase in performance when the exploitation of temporal information was evaluated only in the advantageous zone compared to when it was evaluated in every position of the arena. These results suggest that in the real world, the features detected by the model in the

DBCV are more affected by noise than in the simulated case. For example, since the accuracy of the BDM in the real world experiments is around 80%, and the fact that two views are needed to calculate the DBCV views, the chances of having a good DBCV view are smaller than the chances of having a good SVP view.

7.4.1 Differences between the real world and the simulated case

In these experiments, the conditions differed from those present in the simulated case due to illumination, shadows and reflections, irregularities in the walls of the arena, etc. In the following, I present a brief discussion of these issues and their effects.

The first difference is the process of visual information acquisition. In the simulated experiments, views were collected by the agent while traversing different trajectories. In the experiments presented in this chapter, views were collected by a robot from a grid of positions in the arena. Training views for each strategy are then selected from each image databases. However, although the processes are different, the result is arguably equivalent. A second difference in the visual system is that the cameras in the simulation were ordinary simulated video cameras, whereas in this chapter, a panoramic camera was employed. However, the incoming visual information in both sets of experiments can be considered equivalent because the pre-processing of the panoramic images transformed them into normalised images as in the simulated case.

Another distinction between the real world experiments and the simulated ones is that the output of the blob detection mechanism employed in this chapter differs from the one used in the previous chapters. While in the previous simulated experiments all the selected blobs were objects (or were parts of objects when the agent was too close to the object), in the real world case, in many cases the blobs selected corresponded to artifacts that were not objects. For example reflections, shadows, or simply parts of the arena. Therefore, in this case, the BDM used several criteria to select the detected blobs to be processed. While several blob selection criteria were initially used, the roundness of the blobs was the criteria that showed a better performance. In this case, the selection of the blobs had an accuracy around 80%. Therefore, while the chance of having a bad SVP view would be around 20%, the chance of having a bad DBCV view would be around 40%.

7.4.2 Exploitation of variation in the object views in the real world

The fact that the RBF model can perform object recognition reliably by exploiting simple movement strategies in real world conditions means that the controller required by an agent in order to perform such movements can be simple. It is important to note that the performance of the model is best when the agent is within a particular region in the arena. However, this requirement can be easily fulfilled by using simple object approaching strategies, in which the object is approached first until a threshold in the recognition signal has been reached and, after that, the agent would then perform the movement strategy for view collection. For example, a simple sensory-motor coupling that was tested to provide such simple movement strategies was the use of optic-flow for object approaching (data not shown).

7.5 Conclusion

In this chapter the RBF model was evaluated in the real world was carried out. This chapter was divided in three experiments. In the first experiment, the exploitation of variation in the training views using trajectories was explored. In this experiment it was shown that, as predicted in the simulated case, there are characteristic regions in the arena where the performance of the model is increased when performing recognition tasks. The characteristics of these regions are determined by the variation in the training views which, in turn, are determined by the movements strategies followed by the agent during training. The exploitation of variation while using real world visual information validates the results from the simulated case where the variation in scale and rotation in the training views represented an advantage in the object recognition performance of the model.

The second experiment demonstrated that the temporal information exploitation by the RBF model is not as robust as in the simulated case. However, when the model is restricted to the advantageous zone, the performance of the model suggests that the exploitation of the temporal information is determined by the similarity of the change (in scale and/or rotation) between consecutive views. The results presented in this chapter demonstrated that a simple model can be used to perform object recognition reliably using simple movement strategies in real world conditions.

In the third experiment it was shown that the complexity of the RBF model can be reduced by using an active approach. This approach consisted of actively selecting the training views using an spiral movement strategy. By selecting views based on the neighbourhood similarity, the performance of the RBF model using only a filter size and one orientation was better than when selecting the training views using a fixed interval along the spiral movement strategy.

These experiments validate the results of simulation and show that the RBF model can be used successfully in the real world object recognition.

Chapter 8

Towards active selection of training views

8.1 Introduction

In this chapter, we explore methods of actively selecting training views with the aim of increasing object recognition performance and to assess whether active processes can reduce the need for complex object recognition models.

In previous chapters it was shown that the RBF model could perform as well as a more complex model through the exploitation of active strategies which obviate the need for a hierarchical visual processing system. In this chapter we study this idea further. Specifically, we analyse whether the complexity of the RBF model can be reduced by further exploiting active selection of the training views. This idea is tested here by incrementally reducing the complexity (measured as the number of filters and/or filter types) of the RBF model in such a way that its performance drops and, subsequently, by analysing whether this performance can be regained by the active selection of training views.

We hypothesize that the reduction of RBF complexity (and the attendant reduction of dimensionality of the processed output) will lead to a loss of specificity with respect to the training images acquired during any given behaviour, leading to lower performance during testing (as the processed views of an object might not capture sufficient information about it). To regain performance, it may therefore be useful to select training views with increased specificity by monitoring explicit criteria, applicable during behaviour. That is, selecting views with specific features that make the objects more separable. One simple criterion is *similarity* among training views, as measured by the Euclidean distance between the output vectors from the RBF model. However, from the perspective of object recognition, a set of training views is distinguished not only by specificity of each member with respect to other members, but also with respect to the *representativeness* of each image. By 'representativeness' we refer to the degree to which a given image is representative of images potentially acquired during the training and/or testing phases. Thus, a second criterion to be monitored during behaviour might be a measure of representativeness. However, there is likely a trade-off between representativeness and specificity, analogous to the trade-off between specificity and generalizability common in the machine learning literature (Bishop, 1996). Therefore, the ability of active strategies to improve performance may depend on their ability to select training views which effectively balance this trade-off.

The process of selecting training views for object recognition has been studied previously. In (Mokhtarian and Abbasi, 2005) an automatic method for optimal view selection is proposed based on a similarity measure. However, in their study the object recognition task is not considered for a mobile robot. More recently, Meger et al. (2008) have stressed the importance of autonomous selection of training views in a mobile robot and its implications for cognitive robotics. They use a histogram based method to select angles from which the new training views are collected by an autonomous mobile robot.

In this chapter we explore novel ways of selecting training views with the objective of enhancing their specificity while maintaining representativeness. Specifically, we compare three methods for training views selection. In the first method, which we employ as a control, the training views are selected at fixed intervals along the agent's trajectory. In the second method, which aims to enhance specificity, the views are selected based on a similarity threshold with subsequent views along the trajectory of the agent. In the third method, which aims as well to enhance representativeness, the training views are selected based on their similarity to neighbouring views in the arena. While all these methods are active in the sense that the agent must move to collect the views, there is an increasing level of exploitation of the regularities in the visual information (second method) and the environment through movement (third method).

The sections of this chapter are organised as follows: first I will describe the way in which the RBF model is reduced and show that the object recognition performance of the model, implemented on the gantry robot, decreases as its complexity is reduced (object recognition task and robot are as used in chapter 7). Next, I present a study of the characteristics of the training views collected, focusing on the interactions among specificity, representativeness and performance. These observations guide the development of novel methods for the selection of training views, which are tested using the gantry robot. The chapter concludes with discussion of the results.

8.2 Methods

The experimental setup employed in this chapter is similar to the one used in chapter 7. Briefly, the experiments were carried out using the Gantry robot with a panoramic camera (see section 7.2.1) in an arena of 300cm $\times$ 200cm $\times$ 200cm. Each of seven objects was placed in the center of the arena and images were taken from each vertex of a grid of 290/5 cm $\times$ 170/5 cm positions, except for a 25cm diameter circular buffer region around the centre of the arena where the object was placed. The experiments consisted of two phases. In the first phase, the visual system was trained using the images collected by the gantry robot in certain positions in the arena determined by four movement strategies. This phase was performed for each of the seven objects individually, resulting in seven trained networks. During the second phase, the visual system was tested with object views collected in every valid position in the arena.

The experiments use seven toy-like objects (shown in figure 7.1) labeled: a black truck, frog, ninja, potato man, red truck, squirrel, and truck. The set of images for each object formed an object image database with 2004 images, giving 14, 028 images in total. The images collected using the panoramic camera were "unwrapped" using the algorithm described previously in table 7.1 into 360×400 pixels images. The characteristics of the objects are intentionally chosen so that the recognition tasks were not trivial. For example, there are similar objects in one category (three trucks with similar shapes). Also, the colour intensities of some objects are similar (the potato man and the squirrel).

As with the previous experiments, the visual system employed consisted of the analysis module, classifier module and a blob detection mechanism (BDM) (see sections 3.2.1, 3.2.2 and 3.2.3 for a detailed description). However, while the classifier and the BDM are identical to those used previously, the analysis module consisted of different versions of the RBF model which incrementally reduce its complexity.

8.2.1 RBF versions

The complexity of the RBF model was reduced twice, which resulted in three versions, RBF_A, RBF_B and RBF_C. Each reduction was carried out by decreasing the number of sizes and orientations of the filters used: A) the original RBF model employs four different orientations and four different filter sizes (16 filters); B) For the second version the number of orientations of the filters was decreased to one, 0 degrees and four filter sizes (4 filters); C) Finally, the number of filter sizes was also decreased to one, that is, one filter of size 21×21 oriented at 0 degrees (see table 8.1).

Model	Description	Details
RBF_A	4 orientations 4 sizes	(0^o, 45^o, 90^o, 135^o), (9×9, 11×11, 15×15, 21×21)
RBF_B	1 orientations 4 sizes	(0^o), (9×9, 11×11, 15×15, 21×21)
RBF_C	1 orientations 1 size	(0^o), (21×21)

Table 8.1: Reduction of the RBF model. See text for details.

8.2.2 The classifier module

The classifier employed was the same as described in section 3.2.2. Very briefly, it consists of a set of view tuned units (VTU) used to recognise objects. Each VTU is trained to respond according to the proximity (similarity) between the test view and the training views (see figure 3.12). That is, the more similar the test view to the training views, the stronger the response of the VTU. There is one VTU per object. Each VTU (see figure 3.6) is a set of radial basis functions (or RBF unit). A RBF unit is a Gaussian function G centered on each training view c_i for each object, that is, the centers were located at every training view. The response of each RBF unit is given by:

$$G(c_i, v) = e^{-\|c_i - v\|^2/\sigma_i^2} \tag{8.1}$$

where c_i is the centered-view vector and v is the vector that is being evaluated (test view). Sigma (σ) values are chosen following the optimising method described in section 3.2.2 consisting of exploring values of sigma around $\frac{D}{\sqrt{2M}}$ where D is the maximum distance between RBF training views, and M is the number of centers.

The response y of each VTU for a test vector x is given by

$$y = \sum_i G(v_i, x) W_i \tag{8.2}$$

where G is the activation of each Gaussian function centered on each view of each object. The response of the module y is the linear combination of weights W_i and the Gaussian G. The optimal weights W_i are computed in order to respond with higher values for views that are similar to the views that form a particular VTU, and with lower values for the rest. In section 3.2.2 a detailed description is included about how the VTUs are trained.

Given that the goal of this chapter is to study the conditions in which the performance of the reduced version of RBF can be increased by using an active process in the selection of the training views, instead of optimising σ in the classifier for each version of the RBF model and, in that way, improving the performance of the model, the parameters were only optimised for RBF_A and then used for RBF_B and RBF_C.

8.2.3 Movement strategies

In order to evaluate the performance of the reduced versions of the RBF model and study the characteristics of the training views acquired, we use the four movement strategies previously employed in chapter 7 (see figure 8.1).

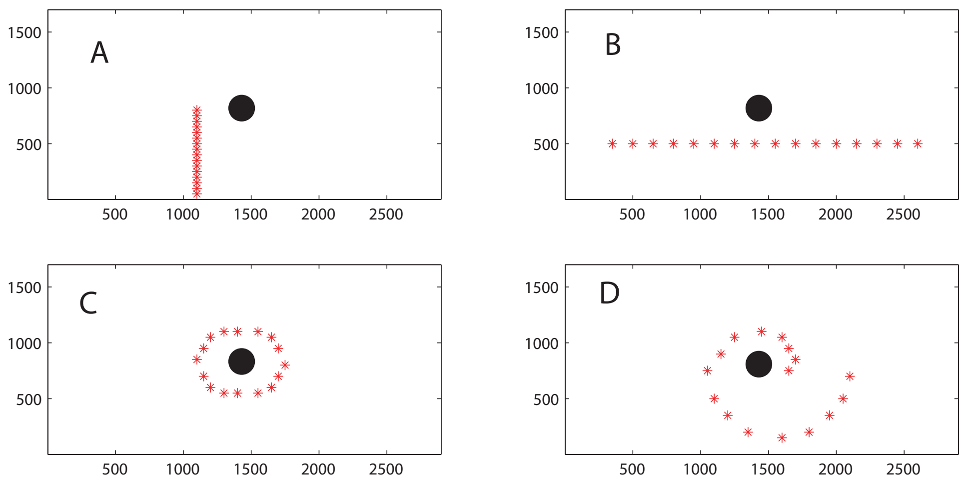

Figure 8.1: Movement strategies used to collect the training views. A) Movement strategy T1: the agent approaches the object in a straight line. B) Movement strategy T2: the agent passes in front of the object in a straight line. C) Movement strategy T3: the agent circles the object. D) Movement strategy T4: the agent spirals around the object.

Each movement strategy provides training views with different characteristics. For example, T1 provides views with limited rotation and scale variance, in comparison with the scale variance of views provided by T2 or rotation variance provided by T3 and T4. We will use these movement strategies to study the impact of the training views on the performance of the reduced versions of the RBF model. This study will allow us to propose methods of selecting training views which increase the performance of the reduced versions of the RBF models by the exploiting regularities in the incoming visual information perceived during agent movement.

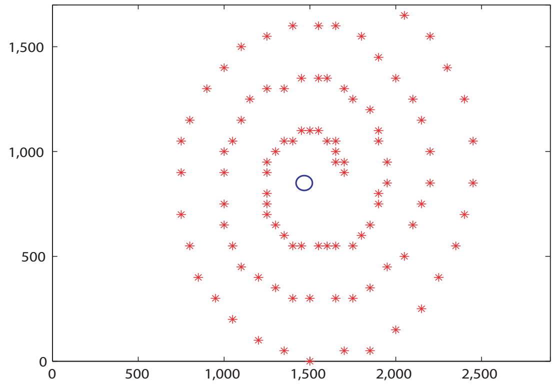

Figure 8.2: Spiral strategy. 100 positions were generated spiraling around the object (circle in the centre).

The methods for training views selection are then tested along the spiral movement strategy (figure 8.2). This movement strategy consists of 100 positions around the objects using an extended version of movement strategy T4 which provides training views with high variation in rotation and scale. The spiral movement strategy is used because, 1) it has a large number of positions (views) to choose from and, 2) it provides significant variation in its views. A more detailed explanation of the methods for selecting training views is given in the following sections.

8.3 Results

The goal in this chapter is to investigate whether the performance of the reduced versions of the RBF model improve when the training views are actively selected (ie exploiting the regularities in the visual information determined by the movement strategies)[1]. The results are organised as follows: first we establish the degree of discrimination performance evoked by the reduction in RBF complexity over four movement strategies and test whether such reduction also leads to reduced specificity of the training views, accounting for the reduction in performance. To rule out trivial explanations of these results, we examine: (i) whether there are particular objects which defeat a particular movement strategy, and (ii) whether the BDM pre-processing affects training view acquisition and hence performance. Next, we study the characteristics of the training views that allow high model performance and, based on this, we propose methods of selecting training views which balance the need for specificity while maintaining representativeness. Finally, a comparison of the performance of the reduced versions of the RBF model and the spatial distribution of the selected views, are presented for each of the new methods of training view selection.

[1]The values of the correct guesses for every RBF version, movement strategy and method for selecting the training views are presented in appendix 9.3.

8.3.1 Reducing the complexity of RBF

The performance of the different versions of the RBF model, RBF_A, RBF_B and RBF_C, was evaluated using movement strategies T1, T2, T3 and T4 (figure 8.1) and shown in figure 8.3. The average performance decreases gradually as the RBF model is reduced. It is likely that the drop in performance is due to the reduction of the number of filters and orientations of the RBF model, which reduces their specificity (or increases their similarity). That is, the views after being processed by the RBF_C look more similar amongst each other than after being processed by RBF_B, and these in turn look more similar than after being processed by RBF_A. This reduction in specificity is illustrated by the similarity map shown in figure 8.4.

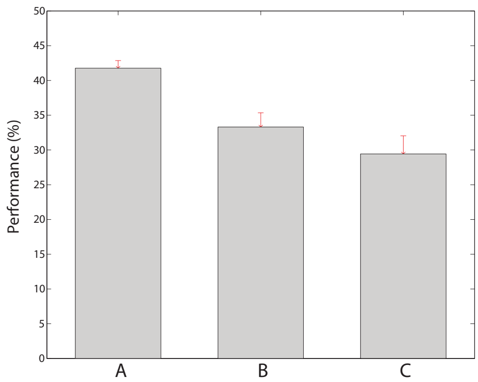

Figure 8.3: Performance (%) of the different RBF reduction implementations. The performance of RBF_A, RBF_B and RBF_C are represented by columns A, B and C, respectively. The performance corresponds to the average number of correct guesses for every position in the arena using the four training movement strategies (averaged across all 4 movement strategies). The error bars show the standard deviation over the movement strategies.

Figure 8.4 shows that the similarity between the training views of each movement strategy increases when using the RBF_C model, in comparison to the RBF_A model. That is, in general, the reddish areas in the RBF_A map turn into yellowish and bluish areas in the RBF_C map. For example, compare the similarity map corresponding to object 4 and strategy 3 of RBF_A and RBF_C (third row, fourth column) in figure 8.4. The similarity map using RBF_B (not shown in this figure) shows an intermediate state between RBF_A and RBF_C.

In some cases, the increase of similarity values between the RBF_A and RBF_C maps are difficult to distinguish. Therefore, we calculated the difference between the RBF_A and RBF_C maps. Figure 8.5 shows the difference of the normalised intensity colour values of the similarity maps for RBF_A and RBF_C. The red colour in this figure represents large positive difference, the white represents zero difference and blue represents negative difference. Clearly, there are a large number of red regions which represent large positive differences, illustrating the reduction of specificity of RBF_C compared to RBF_A.

Additionally, table 8.2 shows the summed difference between the similarity maps for each object and each movement strategy. Positive values represent an increase in the similarity (the views are more similar to each other in the RBF_C map compared to the RBF_A map). These values confirm that (in most cases) the specificity of the RBF model

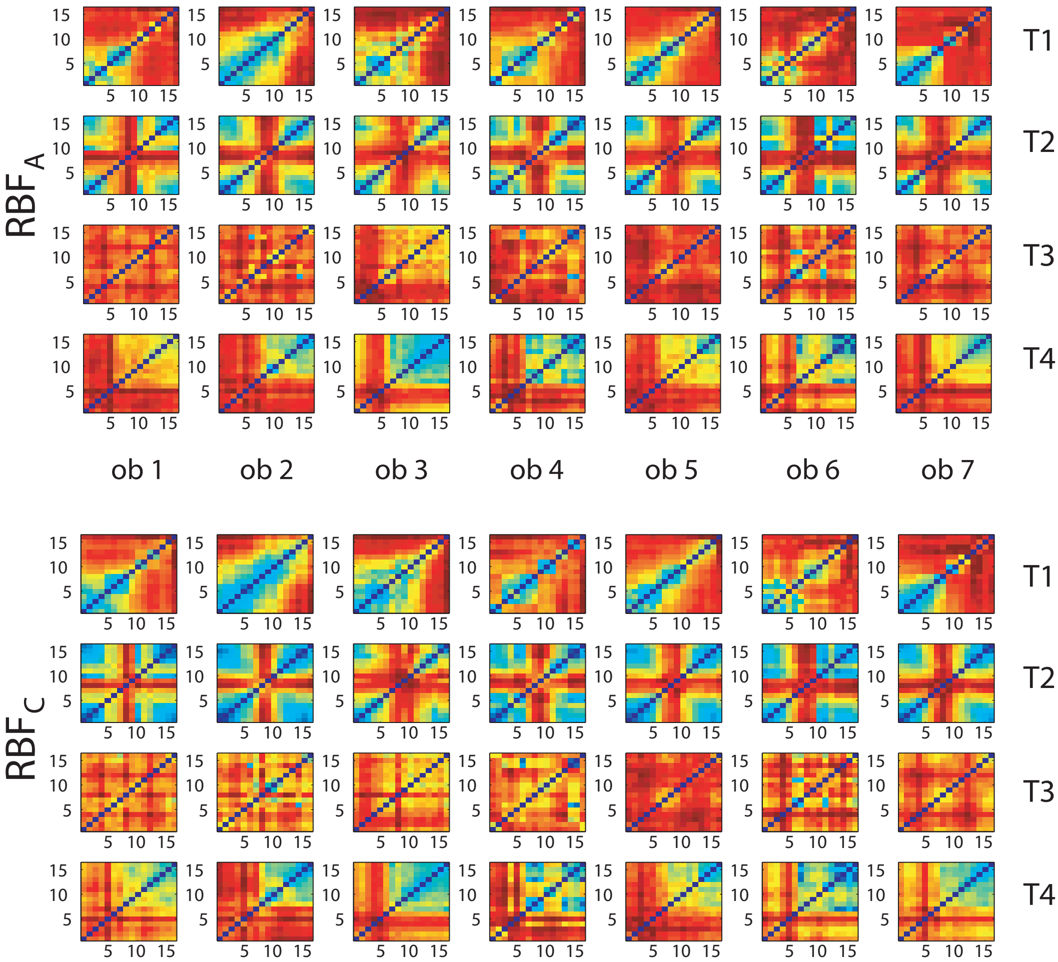

Figure 8.4: Similarity map of the training views using RBF$_A$ and RBF$_C$. The similarity between views v_i and v_j is defined as $\| v_i - v_j \|$ where $\| \cdot \|$ is the Euclidean norm. The red colour in the map indicates the lowest similarity and blue colours indicate the highest similarity between the training views. The red areas in the RBF$_A$ similarity maps are yellowish and blueish in the RBF$_C$, showing the reduction in the specificity in the reduced versions of the RBF model. The similarity map using RBF$_B$ (not shown in this figure) shows an intermediate state between RBF$_A$ and RBF$_C$.

	obj 1	obj 2	obj 3	obj 4	obj 5	obj 6	obj 7
T1	1.039	0.910	1.078	-0.121	0.509	0.849	0.517
T2	1.437	1.414	0.060	0.828	1.202	0.800	1.340
T3	0.040	0.783	0.013	0.441	-0.092	0.447	0.228
T4	0.749	0.448	-0.329	0.006	0.542	0.770	0.755

Table 8.2: Difference between the RBF$_A$ similarity map and RBF$_C$ similarity map. The values are the difference between the sums of the normalised values of each similarity map (for each object and trajectory).

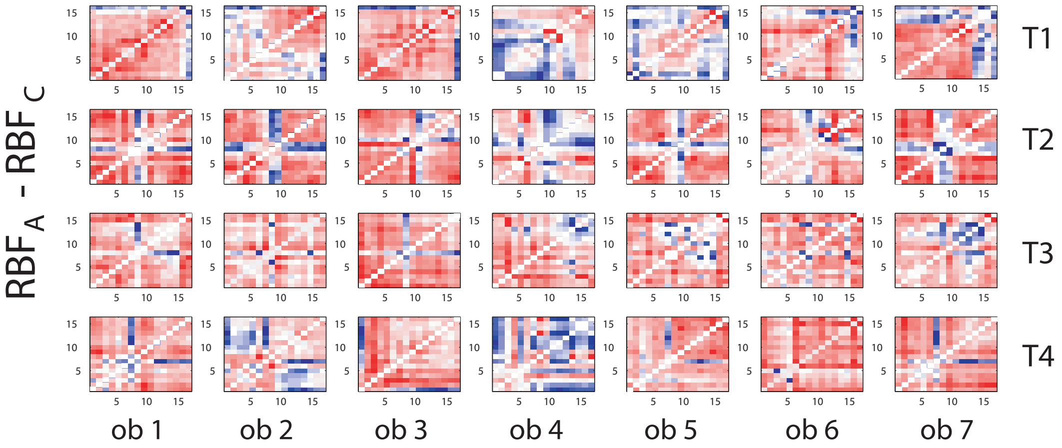

Figure 8.5: Difference of similarity map. In this map, every point represents the difference between the map RBF_A and the map RBF_C after normalising their values by the maximum view difference for each object and movement strategy (from figure 8.4). Blue represents low differences (min value = -0.15), white represents neutral difference (zero) and red represents larger differences (max value = 1.6).

is decreased when its complexity is reduced. This means that it is harder to distinguish between the processed views when RBF_C is used than when RBF_A is used (the RBF_B case would be an intermediate level).

In the next section we will analyse the training views collected using movement strategies T1, T2, T3 and T4, and investigate whether there are measurable characteristics of these views which correlate with increased model performance (in the sense that by using this type of view, the number of test views correctly classified increases).

8.3.2 Investigation of training views and model performance

In order to find a sensible way of actively selecting training views to regain performance it is necessary to find out what makes a good training view, in the sense that these views increase the object recognition performance of the model. To do so, we first compare the performance of the RBF_A model for each movement strategy, and see which one provides better views (figure 8.6). We focus on RBF_A because the classifier widths (σ) were optimised for this version of the RBF model (see section 8.2.2).

Here we see that the performance of strategy T2 is the best, followed by strategy T4, then T1 and T3 being clearly the worst. Before studying the characteristics of the training views for each movement strategy, we determine if the model performance is affected by other factors.

Are there particular objects that affect the models' performance?

In general, the performance of a particular movement strategy could depend on the intrinsic properties of the objects or the training views collected. That is, there are particular objects that are difficult to recognise using a movement strategy, or the movement strategies provide training views that contain features that allow the models to classify the objects correctly.

Examining the performance of the movement strategies for each object (Figure 8.7) we

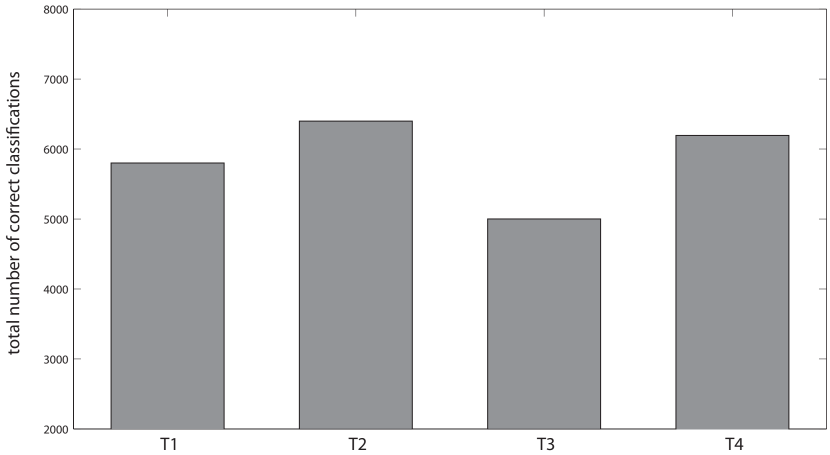

Figure 8.6: Total number of correct classifications by the RBF_A when the training views were collected using movement strategies T1, T2, T3 and T4 across all objects.

see that there are objects (object 2 and 6) that are difficult to recognise for every movement strategy in comparison with the others. However, as we also see in figure 8.7, there is no clear evidence that a particular movement strategy confers any advantage when dealing with such objects (in the sense the number of correct classifications for any strategy is small). Therefore, we assume good performance means that, in some sense, the positions in the movement strategy are providing good training views.

However, given that before being processed by the RBF models, the blobs, which in turn determine the views, are detected (correctly or not) by the BDM, we need to analyse whether this pre-processing has an impact on which movement strategy provides better views.

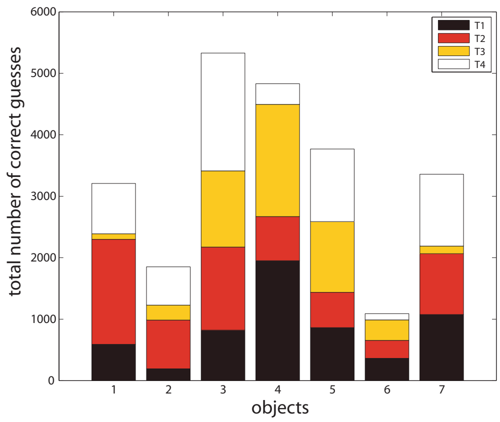

Figure 8.7: Total number of correct guesses for each object for the RBF_A model when using movement strategies T1, T2, T3 and T4. Objects 2 and 6 are the ones with the lowest performance. Note that the quantities represented by each colour are independent and they are not meant to represent a cumulative plot.

Does the BDM affect the movement strategies' performance?

The BDM could affect the performance of the models when, for example, an object's colour makes it difficult for the BDM to detect it when viewed from a certain angle. As seen in the case of the squirrel object (object 6), the colour of this object makes the BDM struggle to detect "good blobs" for some movement strategies (figure 8.8). In general, a correct blob was chosen when the view contained most of the object. In contrast, a "bad" blob is when the BDM selected a view which contains something that is not an object or only a small part of it. Examining the number of bad blobs returned by the BDM for all objects (Table 8.3), we see that the movement strategies do differ in how many bad blobs they return, but that this number is not correlated with object recognition performance.

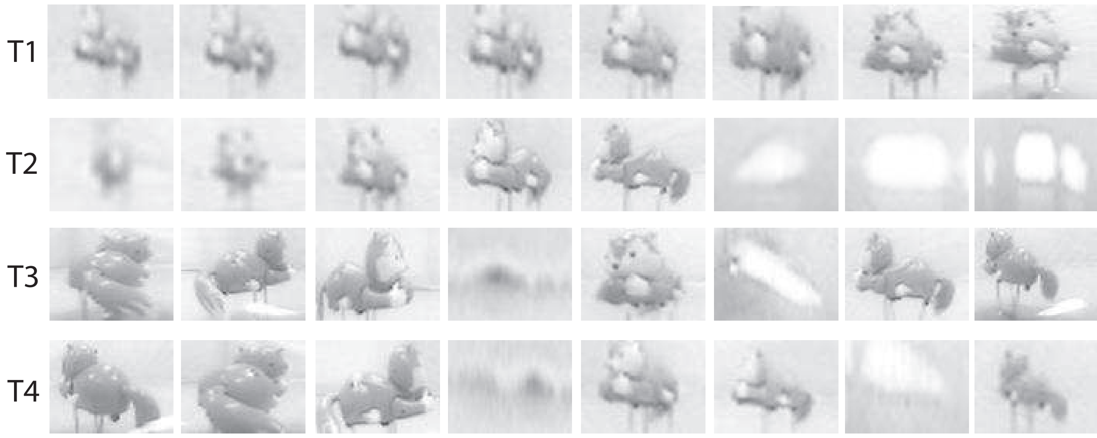

Figure 8.8: Examples of training views of object 4 (squirrel) using movement strategies T1, T2, T3 and T4.

Object	T1	T2	T3	T4
1	0	1	0	0
2	0	1	2	0
3	0	2	0	1
4	0	3	3	2
5	0	1	1	0
6	0	6	3	4
7	0	0	0	0
total	0	14	9	7

Table 8.3: Number of "bad" blobs in the movement strategies. Movement strategy T2 has the largest number of bad blobs amongst all the movement strategies. Object 6 is wrongly detected the largest number of times.

To fully assess the impact of bad blobs returned by the BDM on the recognition performance of the movement strategies, the bad blobs detected were replaced with manually selected good blobs. The blob corrections consisted of replacing reflections for objects, and replacing part of an object with a more complete object. Figure 8.9 shows the total number of correct classifications for each object and each movement strategy for the RBF_A model when the 'errors' of the BDM were corrected. Note that, even though the number of correct guesses generally increases (mainly for object 4), objects 2 and 6 are again the hardest to classify. In addition, there is no evident advantage of T2 and T4 over the rest of the movement strategies. Finally, note that there are objects for which the general performance decreases when the blobs are corrected (objects 3 and 5). This is because the blobs detected by the BDM at locations where a bad blob was detected, are sometimes also bad blobs. Thus, such test views are more similar to a bad blob, than to a corrected blob, and in this case, having a bad blob as a training view can be advantageous as it better represents the test views.

Overall, however, after correcting the BDM, the performance of RBF_A is broadly (Figure 8.10) similar and, therefore, we conclude that the BDM does not play a significant role in the way the training views affect the performance of the RBF models. Hereafter, the experiments are carried out using the BDM detected blobs without manual corrections.

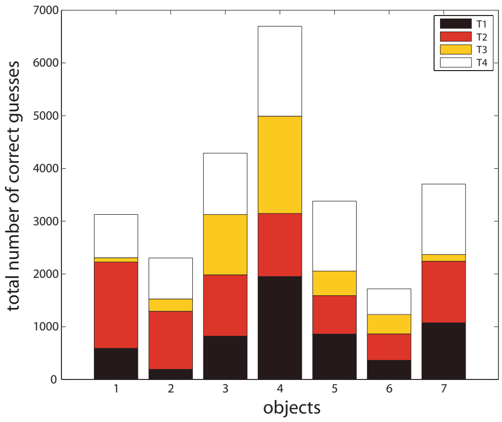

Figure 8.9: Total number of correct guesses for each object and each movement strategy (T1, T2, T3 and T4) for the RBF_A model. Note that the quantities represented by each colour are independent and they are not meant to represent a cumulative plot.

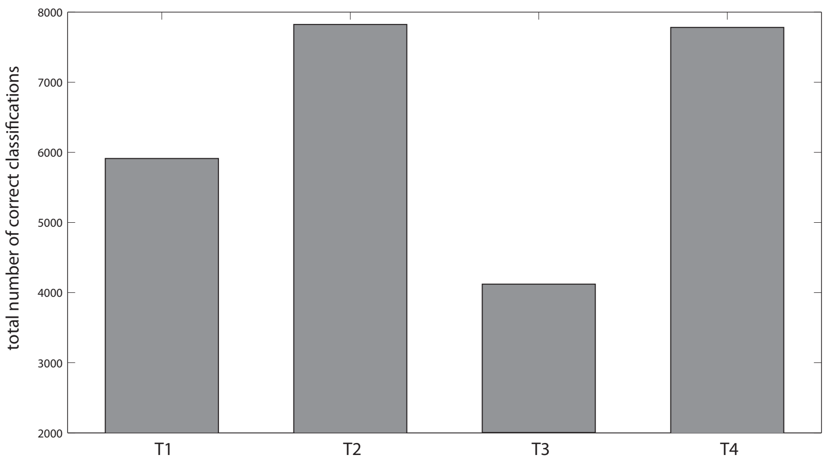

Figure 8.10: Total number of correct classifications by the RBF_A model using movement strategies T1, T2, T3 and T4, when the blobs were manually corrected.

Knowing that neither particular objects nor the errors of the BDM play a major role in the performance of the RBF models when using the different movement strategies, we now analyse the specificity and representativeness properties of the training views in the movement strategies.

Specificity and representation of the training views

First, we evaluate whether the similarity between the training views determines which movement strategy provides better views. As the decrease in RBF model performance correlates with an increase in the similarity (as the specificity decreases from RBF_A to RBF_C), we might expect that the similarity of the views collected by both movement strategies, T2 and T4, would be lower than both movement strategies T1 and T3. However, the distances between the training views for movement strategies T1, T2, T3 and T4, as seen in table 8.4, show that T1 and T2 are the movement strategies with smaller differences (higher similarities) and, T3 and T4 with larger differences (lowest similarities) for RBF_A, RBF_B and RBF_C.

Strategy	RBF_A	RBF_B	RBF_C	total
T1	**3960.3**	1950.1	488.4	6398.8
T2	**3862.7**	1922.8	483.4	6268.9
T3	*5625.9*	2879.7	756.7	9262.3
T4	*5115.7*	2547.6	668.6	8331.9

Table 8.4: Total sum of the distances between the training views of the movement strategies for the RBF versions. For the three versions of RBF movement strategies T1 and T2 show lower distances compared with the distances for movement strategies T3 and T4. The distance values of the strategies for RBF_A are in bold and the largest distance values are in italics.

Therefore, the similarity amongst the training views is not enough to explain why the training views collected using one movement strategy are better than when using another strategy. Intuitively, this makes sense because in general, we want training views to be different to each other so that the system can account for variation in object appearance. However, if we select training views that are as dissimilar as possible, we may end up with a training view that does not look like any test view of the object. We therefore also need to consider training views that are representatives of the test views.

One way in which a training view represents a set of test views is that it is taken at a similar spatial distance from the object in the arena. This is particularly important here as the spatial distance of the training views could influence the performance of the strategies due to the resizing process in the BDM. For example, views that are collected far away from the object become 'averaged' (blurred) by the BDM. Therefore, these views are very similar. In contrast, views that are collected close to the object keep their 'detail' (distinctive features) and are very different to each other.

By measuring the spatial distance between the locations in the arena where the views were collected, we note that the distribution of the spatial distances is different for each movement strategy (figure 8.11). The standard deviation of the distance between the views and the object broadly correlates with performance ($\text{std}_{T1} = 172.63$, $\text{std}_{T2} = 287.20$, $\text{std}_{T3} = 29.19$, $\text{std}_{T4} = 208.75$).

The influence of the positions of the training views can be confirmed by figure 8.11 which shows the distribution of correct classification in the arena and the locations where the training views were collected. The spatial distribution of the training views (white dots) around the arena is topographically correlated with performance. When the training views are collected close to the object only (movement strategy 3), the correct guesses are mostly located close to the object. In contrast, when the training views cover a wider area (movement strategy 2 and 4), the correct guesses are located over a wider region

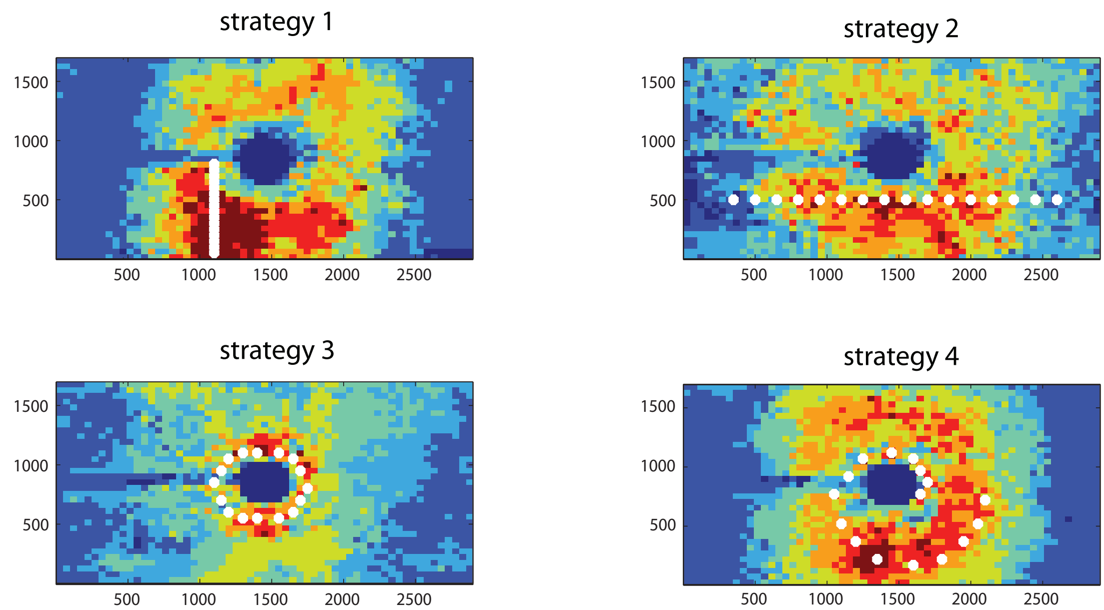

Figure 8.11: Map of correct guesses for each movement strategy. The darkest blue colour represents 0 correct classifications and the darkest red colour represents 7 correct classifications at a given location in the arena (one per object). The white points represent the locations in the arena where the training views were collected for each movement strategy.

in the arena. Intuitively, when the training views are close to the object, they are more specific since they have more detail of the objects in comparison with most of the views in the arena. In contrast, when the training views are taken from larger distances to the object, the views are more general, so that the training views are similar to the majority of the views. Additionally, we observe that in movement strategies 2 and 4, the views are collected not only far away from the object, but also close to the object. This is important because when the views are taken from a short distance to the object its features are captured so it can be discriminated from the rest of the objects.

Summarising, a good training view is one that is a good representative of a group of views in the arena around the position where the training view was taken but, at the same time, it offers distinctive features that allow the model to distinguish this view from other objects. Now we will define different ways of selecting training views that attempt to capture these features and evaluate if actively selecting good training views (in this sense) increases the performance of reduced versions of the RBF model.

8.3.3 Exploiting regularities in the environment through movement

In order to evaluate new training view selection methods, we need first to establish a baseline of "good" performance. To do this, we use the interval based selection method. This method consists of choosing the training views from the positions along the spiral movement strategy in fixed intervals in two directions, normal and inverse order (figure 8.12). For example, for an interval of 1 view, the training views would be collected using positions 1, 3, 5, ..., 29, 31. When the order is inverse, the views are selected starting from position 100 in the spiral movement strategy (which is the furthest position from the object). For example, for an interval of 6 views with inverse order, the training views would be collected from positions 100, 94, 88, ..., 16, 10 (figure 8.13). We used intervals of 1 - 6 views to select the training views for normal and inverse order. The largest interval in order to have 16 views from the 100 available positions was 6.

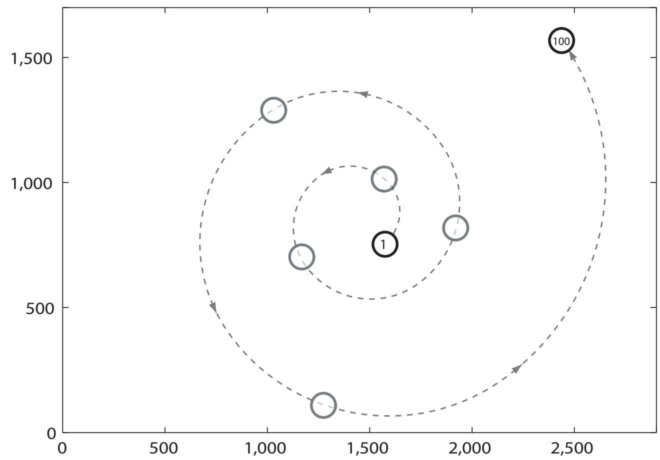

Figure 8.12: Order of the views along the spiral movement strategy. The circles represent positions along the spiral trajectory (dotted line) where views were collected. The normal order is anticlock wise, the first view (1st) is the one closer to the object and the last one (100th) is the one further away from the object. The inverse order is in the opposite direction, clock wise, starting with the further position in the trajectory from the object.

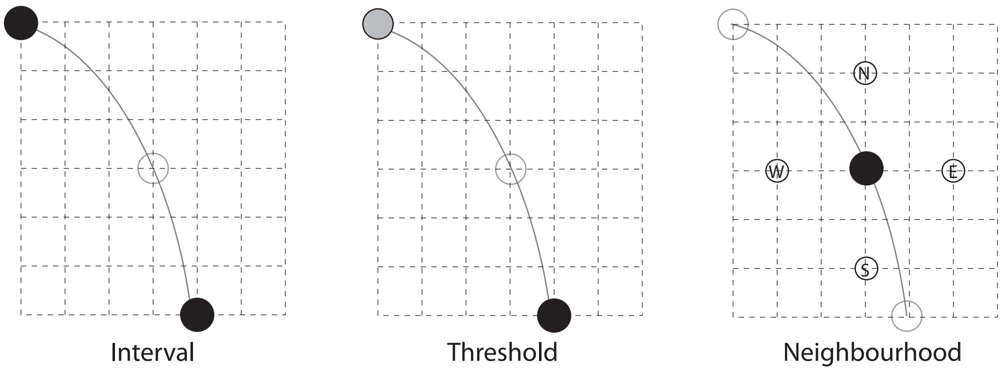

Figure 8.13: Examples of the training views selection methods. The dotted lines represent a region in the arena and the intersections of these lines represent a valid position in the arena. The solid line represents a segment of the trajectory of the spiral movement strategy. The interval based method (left) selects the training views at fixed intervals. The solid circles represent positions selected to collect training views and non-solid circle represents a non-selected position (interval 1 in this case). The threshold based method (middle) measures the similarity between the current view (black solid circle) and the next view (grey solid circle). The non-solid circle represents a previous point in the spiral where the similarity was not lower than the threshold. Finally, the neighbourhood based method (right) calculates the similarity values between the current view in the spiral movement strategy and its neighbours (in the arena) in the four cardinal directions.

The performance of RBF_A, RBF_B and RBF_C using different intervals is shown in figure 8.14. We observe that for RBF_A, RBF_B, and RBF_C, using large intervals to select the training views is better than using small intervals, except when the training views are selected from the spiral strategy in inverse order. This would suggest that the first views in the spiral movement strategies are not good training views because, the 'best' intervals select less of these views.

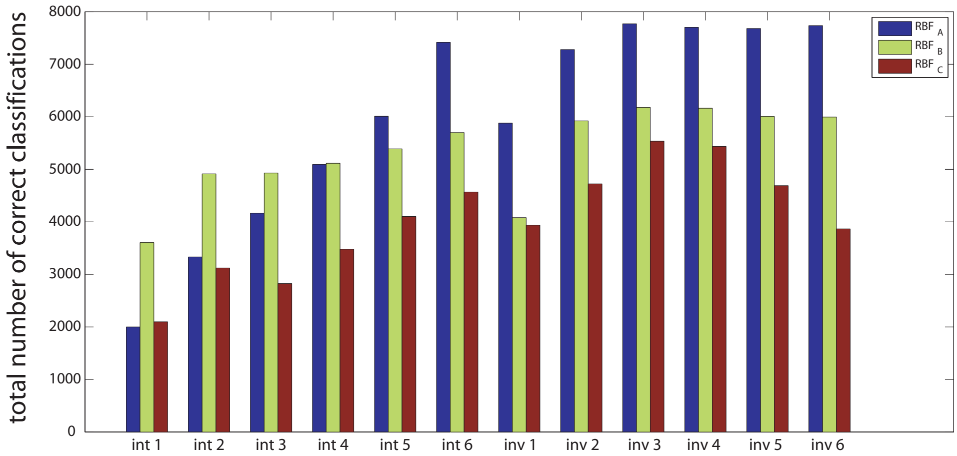

Figure 8.14: Total number of correct classifications by the RBF models when the training views were selected using a fixed interval strategy. Six intervals were used in both, normal and inverse order for each RBF version. The same classifier parameters ($\sigma = 1.8$) were used for RBF$_B$ and RBF$_C$ not only for this training view selection method but also for the threshold and neighbourhood based methods as well.

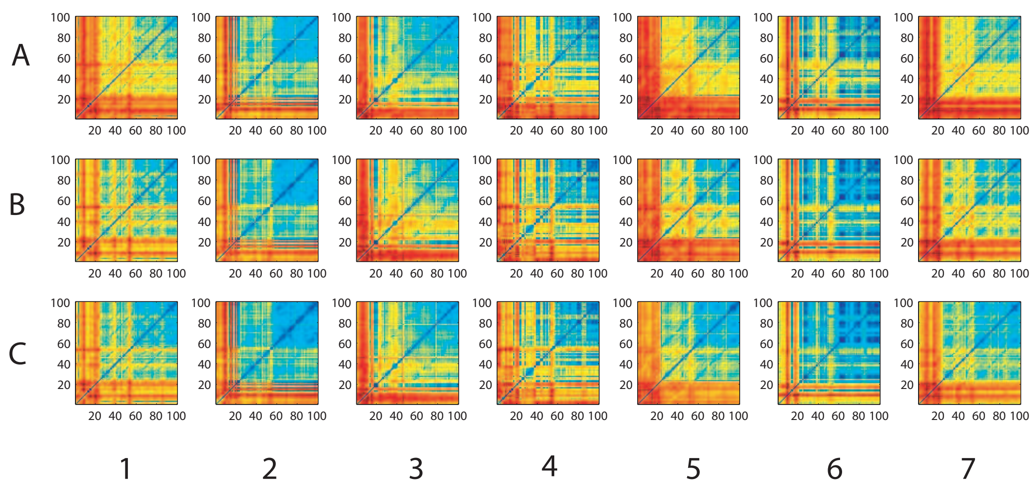

Figure 8.15: Similarity map of the views in the extended spiral movement strategy for the 7 objects (columns) and the RBF$_A$, RBF$_B$ and RBF$_C$ models (rows A, B and C respectively). Red colour represents low similarity between the views and blue colour represents high similarity between the views.

This intuition is reinforced by a similarity map of the training views where we see that, approximately, the first 25 views in the spiral strategy are different to the rest and each other (observe the reddish L-shape region at the bottom rows and the columns at the left of the similarity maps in figure 8.15). These views are unlikely to be good training views because they are not similar to the rest of the views, and so they might not 'represent' any group of views. Thus, using larger intervals may be beneficial as it avoids most of the first 25 views. Similarly, by choosing training views with intervals in inverse order (starting at the furthest view from the object, which is view number 100) we avoid the first 25 views. In general, the best results were found when an interval of 3 views in inverse order was used.

Of course, there is no way of knowing in advance where this set of problematic views are. Thus, there is no way of telling what interval is best and in which order the views should be selected without knowing the correct classification outcome. What we therefore want is some way of autonomously selecting training views along a particular trajectory based on the properties of the training views.

Threshold based training views selection

One way of selecting training views is to take a more active approach, in the sense that the system can control (to some degree) which views are selected, by maintaining some criteria on the incoming visual information (rather than selecting training views at fixed intervals). We want a method which selects views following the criterion of keeping a degree of dissimilarity between the training views so the system could account for variation in the test views but, at the same time, we want views that are still good representatives, unlike the first 25 views along the spiral (see figure 8.15). Therefore, we want to select views that are different to each other but not too much, so we use a threshold.

This strategy of selecting training views consisted of keeping track of the distance between the current and subsequent training view along the spiral strategy:

1. Measure the Euclidean distance between the current and the next training view (along the spiral).
2. If the distance between the current and the next training view was smaller than the threshold, the current view was kept as a training view, the threshold was updated and the current view was moved further.
3. Otherwise, the next view was moved further along the spiral strategy, the threshold was updated, and the distance was taken again (start from 1).

This process was repeated until sixteen training views were collected (figure 8.13).

The threshold was initially chosen as the average of the Euclidean distances between all the views in the spiral movement strategy. However, since the distances between views processed by each RBF version are different, that is $\| \mathrm{RBF}_A(v) \| \geq \| \mathrm{RBF}_B(v) \| \geq \| \mathrm{RBF}_C(v) \|$, and also the distances amongst the training views are different for each object, the thresholds were different for every version of RBF model and for every object (table 8.5).

More troublingly, the distances between the training views change considerably along the spiral (ie the distance between the initial views is larger than between the views at the end of the spiral), therefore, the threshold needed to be adaptive to reflect this. Otherwise, once the current view passed the first views which are very dissimilar, only the immediate subsequent views would be selected. Therefore, the threshold was updated in two ways: (1) Every time that a subsequent view was not selected, the threshold was increased by a constant step (S_m) in order to account for the variation in the differences between the views along the spiral for every model m. That is, $T_{m,o} = T_{m,o} + S_m$, where $T_{m,o}$ is the threshold for the model m (RBF$_A$, RBF$_B$ or RBF$_C$), and object o; (2) When a view was selected, the threshold was equal to the distance between the selected view and the current view.

model	obj 1	obj 2	obj 3	obj 4	obj 5	obj 6	obj 7
RBF$_A$	8.22	3.94	10.76	5.76	7.25	3.18	7.36
RBF$_B$	3.76	2.29	3.23	2.94	2.82	1.95	3.19
RBF$_C$	1.45	1.29	1.33	1.32	1.36	1.19	1.42

Table 8.5: Thresholds for every object and every RBF version.

Table 8.5 shows the threshold for every version of the RBF model and every object. The proportional steps (S_m) for the RBF$_A$, RBF$_B$ and RBF$_C$ were 0.0127, 0.0106 and 0.0067 respectively, and they are calculated as (an approximation of) the average of the variation in the similarity between consecutive views for all the objects and each model.

Table 8.6 shows the performance (total number of correct classifications) for the three versions of the RBF model when using the threshold based method to select training views in the spiral movement strategy.

RBF$_A$	RBF$_B$	RBF$_C$
6021	5985	3147

Table 8.6: Total number of correct guesses by the RBF models when using the threshold based method to collect the training views.

This method presents an improvement in the performance of the RBF$_A$ model over the interval based method for intervals of 1, 2, 3, and 4 in normal order and interval 1 in inverse order. Analogously for RBF$_B$, the threshold method outperforms the interval based method for intervals of 1 - 6 views in normal order and also when using an interval of 1 and 6 views in inverse order. Finally, for the RBF$_C$, the threshold based method presents an increase in performance over the interval based method for intervals 1, 2, and 3 in normal order (not for intervals in inverse order). However, our aim is to regain performance in the reduced versions of the RBF model in comparison with the complete version (RBF$_A$). Even though this happens to some extent for the RBF$_B$ model (its performance is close to the performance of the RBF$_A$), the performance of RBF$_C$ does not increase as much. In particular, the performance of RBF$_C$ does not show an improvement when using the threshold based method, compared to the interval based method with interval larger than 1. Therefore, we conclude that the views selected using this method still lack some characteristics in order to regain the performance lost by the reduction in complexity (especially for the most reduced version of the RBF).

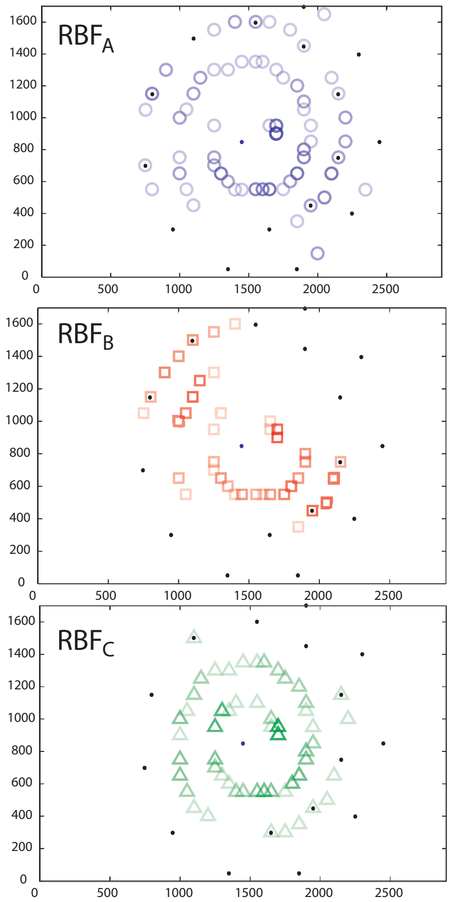

Figure 8.16: Positions where the training views were collected in the arena when using the threshold based method and the RBF$_A$, RBF$_B$ and RBF$_C$ models for all objects. The intensity of the colour of blue circles (RBF$_A$), red squares (RBF$_B$) and green triangles (RBF$_C$) represents the number of views selected in that position in the arena. The more intense the colour, the more views were selected in that position. The black dots represent the positions of the views using an interval of 3 views in inverse order.

Looking at the spatial distribution of the locations where the views were selected by the threshold method, we observe that the views selected using RBF$_A$ are distributed further away from the object than views selected using RBF$_B$ which, in turn, are distributed further away from the object than the views selected using RBF$_C$ (figure 8.16). Thus, the less complex the model, the closer the training views are to the object. This might be because views that are closer to the object contain more detail than the views that are collected away from the object. However, this does not seem to help, at least for the RBF$_C$, as the performance is fairly poor in comparison with RBF$_B$ (and RBF$_A$).

Thus, it seems that using only a measure of the similarity between the views along the spiral is not sufficient to regain the performance lost by the reduction in the complexity of the models (especially for the simplest version of the RBF). More over, we also observe that, while this selection method avoids selecting most of the training views that are closest to the object, it selects several views which are close to each other. This results in having views that are not good representatives, in the sense that we would like to have views that represent the largest number of views in the arena (see section 8.3.2).

Neighbourhood based training views selection

Following the idea that a representative training view is one that is similar to its surrounding views, we propose a method for selecting views based on their similarity with the views in a surrounding area or neighbourhood in the arena. We define the neighbourhood similarity $N_{sim}(v)$ of a view v as the sum of the similarities between v and its four cardinal neighbours:

$$N_{sim}(v) = \sum_{i=1}^{4} \parallel v - n_i \parallel \tag{8.3}$$

where v is the current view in the spiral strategy, $\parallel \cdot \parallel$ is the Euclidian norm, and n_i is one of the four neighbours of v in the cardinal directions with 2 positions of spatial distance in the arena, between n_i and v (figure 8.13). Different spatial distances were tested for the neighbourhood of the current view, but a (radial) distance of 2 had the best results. In order to select the training views, N_{sim} was calculated at each point along the spiral strategy and the 16 views with the lowest N_{sim} values were selected. Note that, unlike the threshold method which measures the similarity between views only along the spiral, the neighbourhood method exploits regularities in the visual environment more explicitly by measuring the similarity in the surrounding views within the arena. We also tried to only measure similarity for the positions along the spiral movement strategy but better results were found if the similarity was considered in the arena (neighbourhood) rather than along the spiral.

RBF$_A$	RBF$_B$	RBF$_C$
7648	6152	5604

Table 8.7: Total number of correct classifications by the RBF models when using the neighbourhood based method to collect the training views.

Table 8.7 shows the performance for each RBF version when using the neighbourhood method to select the training views. To compare the performance of the models using the selection methods, figure 8.17 shows the total number of correct classifications of the RBF versions when using the fixed interval, threshold and neighbourhood based methods to select training views along the spiral movement strategy. For the fixed interval method (first two groups of bars), only the worst and the best interval values are shown (interval of 1 view in normal order, and interval of 3 views in inverse order respectively). Firstly, note that the performance of the models when using the neighbourhood method is generally better than when the views were selected using the threshold method. Additionally, observe that with this method all the models match the performance of the best set of views for each model derived from the interval base method (interval 3 inverse order and neighbourhood in figure 8.17).

In order to better understand the differences in performance for the training view selection methods, we compare the locations where the training views were collected using

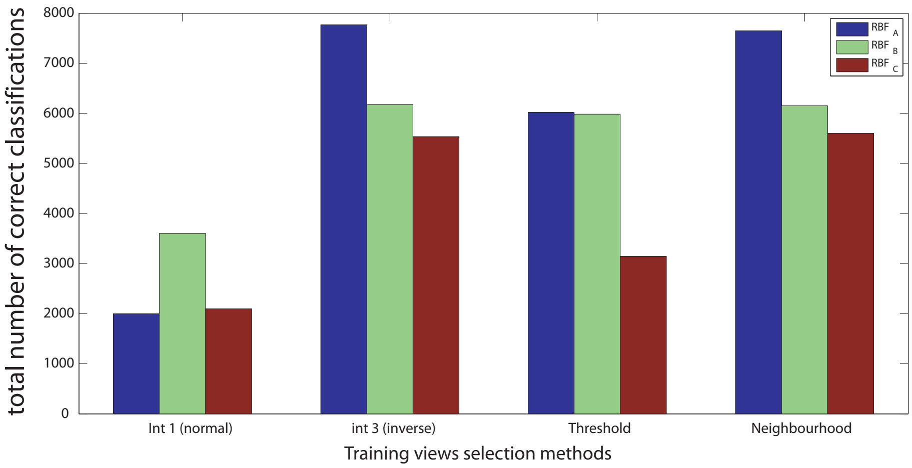

Figure 8.17: Total number of correct classifications for RBF_A, RBF_B and RBF_C models when different methods for selecting training views were employed.

the different selection methods (figure 8.18). Note that the views selected using the interval

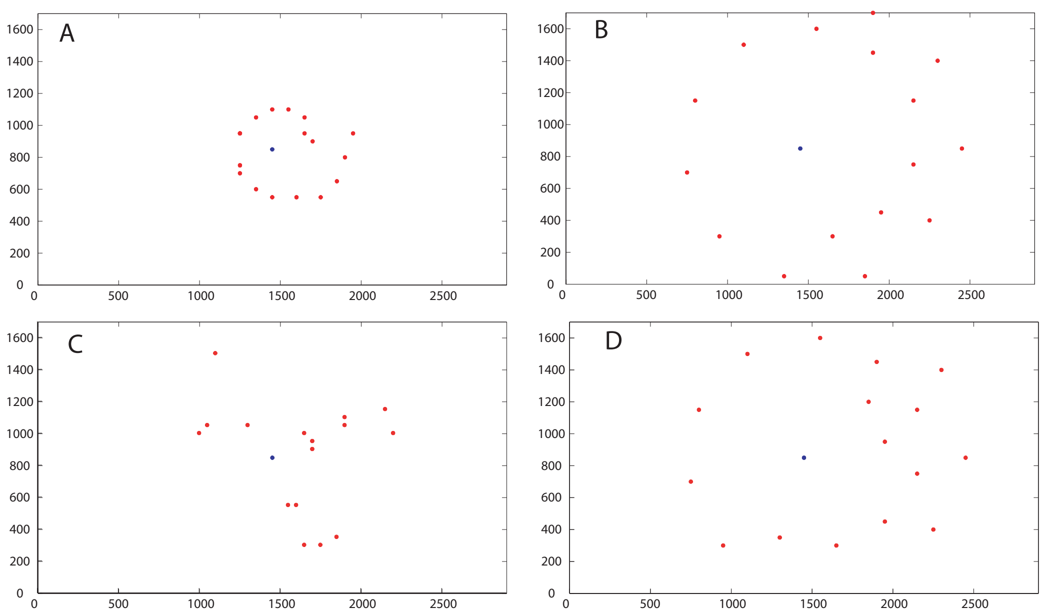

Figure 8.18: Example of locations where the training views were collected for object 1 using the RBF_A model for the different view selection methods. Red dots represent the locations where the views were collected and the blue dots represent the object location in the arena. (A) Interval based method with interval of 1 view in normal order. (B) Interval based method with interval of 3 and inverse order. (C) Threshold based method. (D) Neighbourhood based method.

based method with interval of 1 view (A), are similar to movement strategy 3 (which had poor performance). Also note that the optimal set of views, which is the interval based method with interval of 3 views in inverse order (B), is similar to the set of views selected by the neighbourhood based method (D). Therefore, we can see that what the best set of views (B), and the neighourhood selected views (D), have in common is that they are spread out (ie they are good representatives of test views). However, are there differences between the distances of the views when using different versions of the model when employing the neighbourhood method? In contrast with distribution of the training views selected by the threshold based method, the positions of the views selected using the neighbourhood similarity are roughly the same for every model (figure 8.19).

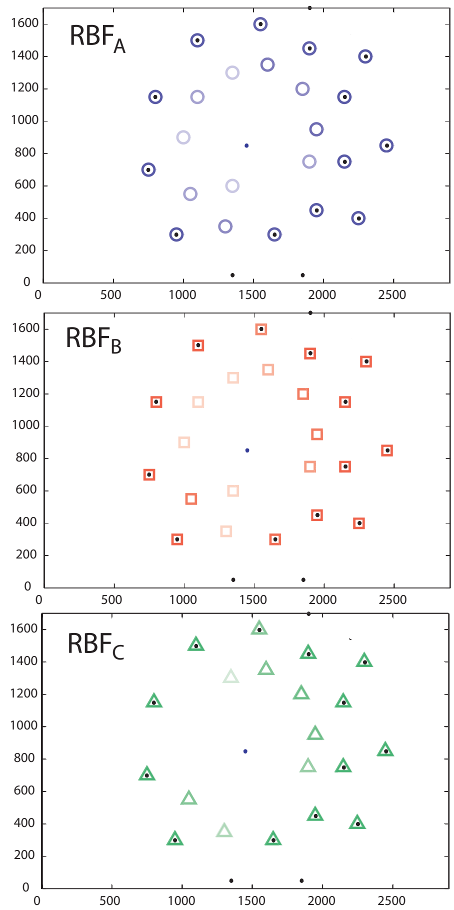

Figure 8.19: Positions where training views were collected using the neighbourhood based method for the three versions of the RBF model and every object. The intensity of the colour of blue circles (RBF$_A$), red squares (RBF$_B$) and green triangles (RBF$_C$) represents the number of views selected in that position in the arena. The more intense the colour, the more views were selected in that position. The black dots represent the positions of the views using an interval of 3 views in inverse order.

More over, even though this method selected better views than when using the threshold method, neither the performance of the RBF$_B$ nor RBF$_C$ matched the performance of the more complex model (RBF$_A$ and RBF$_B$ respectively) when the views were selected using the neighbourhood method. However, the performance of the RBF$_C$ model is closer to the RBF$_B$ performance in this case, compared to the threshold based method (compare brown and green bars for the fourth group of bars in figure 8.17). Therefore, we can conclude that even though the performance of the models when using the neighbourhood method matches the performance of the models when the best set of training views at regular intervals is used, the former method does not select views that contain features that increase object separability to the point that reduced versions of the RBF model match the performance of more complex RBF versions.

8.4 Discussion

8.4.1 How could the reduced versions of the RBF model fully regain performance?

The performance of reduced versions of the RBF model rely on two factors: 1) specificity in the views, that is, features that make the processed views of different objects separable and, 2) having representative views, that is, views that represent a group of potential views in the arena. However, these two factors can counteract each other. That is, a very specific view can be a bad representative, in the sense that its neighbours are not similar to itself. For example, the views that are very close to the object in the arena are very specific but they are not similar to any of the other object views. Therefore, an optimal balance of these two factors could be important to regain performance, meaning that, the specificity lost by the complexity reduction can be regained by exploiting regularities in the incoming visual information in order to select training views with high specificity but maintaining their representativeness.

However, exploiting regularities in the visual information to select training views as presented in this work and achieving a balance of specificity and representativeness is not a trivial task, at least not in our system. That is, in order to know exactly how much specificity is required in the training views, it is necessary to evaluate their separability with respect to the other object views, which is not possible in our mutually exclusive training-testing phase paradigm.

The fact that even with the best set of training views at fixed intervals (control) it was not possible to match the performance of the more complex versions of the RBF model, suggests that training view selection on its own can not account for the complexity reduction. That is, it may not be such views in the arena that would make the reduced versions of the RBF model regain performance completely. This might be because the features used by the more complex versions of RBF to separate the object views do not depend on the spatial distribution of the views but on the nature of the incoming visual information.

A possible solution for this problem could be to manipulate the incoming visual information so that the features detected by the reduced versions of the model are sufficient to match the performance of the more complex models. For instance, rotation of the views so the response of the filters of the reduced RBF versions is enhanced in different ways for each object. However, even this was possible, useful criteria to select such views, such as the ones presented in this work, would be needed.

8.4.2 Towards active object recognition

We note that selection of training views is not the only point in the object recognition process where regularities within the environment could be exploited by adaptive or active processes. For example, the classifier parameter σ could be adaptively set to make the VTU units more or less specific in their responses according to variability in object views. Alternatively, the path the agent takes during training and/or testing could be guided by properties of the incoming views. However, such alternatives may well require criteria similar to those that we explore here, such as representativeness and similarity, to enable them to function. Moreover, these examples require a complex study of the relationship between the features detected by the models for each object and the classifier (for instance σ would be highly dependent on each object) or a complex study of the relationship between motor control signals and the response of the models. Thus, we focus on training view selection here and leave the alternatives mentioned as possible lines of future investigation which might build on the work presented here.

8.5 Conclusions

The idea of this chapter was to explore methods of active selection of training views and to assess whether strategies that exploit regularities in the visual input made apparent through agent movement can compensate for reduced complexity in the object recognition model.

To this end, we incrementally reduced the complexity of the RBF model, by reducing the number and types of its filters, and demonstrated that this reduction, which diminishes the specificity of processed images, also reduces its performance. By examining the training views provided by four movement strategies, we argued that accurate object recognition requires the selection of training views that are distinct, so they can account for variation in object appearance, whilst also being representative of the test views. That is, training views should balance the competing pulls of specificity and representativeness.

We therefore designed two methods of training view selection, the threshold and neighbourhood method, which measure the similarity and representativeness respectively, of processed images during a spiral movement around the object to be recognised. The threshold method attempted to select sufficiently distinct training views by measuring the variation in similarity of the views along the spiral. The neighbourhood method aimed for representative training views by exploiting environment regularity more explicitly and measuring how well candidate training views in the spiral represented their surroundings. We then tested these methods using the original and reduced RBF models and compared their performance to a control training view selection method, the interval method, which selected views at fixed intervals along the spiral and was thus independent of the visual system employed or the characteristics of the views in the arena.

The results show that by exploiting the regularities in the visual information through movement, the minimal RBF model (RBF_C) could not regain 100% of the performance of the complete version (RBF_A) (which was lost with the reduction of the RBF model). However, it was demonstrated that by using the neighbourhood based method to select training views, the performance of the minimal RBF (RBF_C) can be as high as when using the best set of views selected at fixed intervals based method (interval of 3 views in inverse order) along the spiral movement strategy (see figure 8.17). Additionally, it was also shown that the performance of the RBF_C using the neighbourhood method is close to the performance of the RBF_B using the best set of training views (selected using the fixed interval based method). Therefore, we can conclude that, exploiting the regularities in the visual information using movement proved to be, at least, a good alternative as a criteria to select training views in a mobile agent.

This chapter stresses the fact that the performance of a visual system not only depends on its complexity, but also on the exploitation of its ability to move within the environment. We have shown that by exploiting movement during training and tracking criteria which can be measured during behaviour, the performance of RBF models can be improved. In summary, we posit that the further development of active strategies for object recognition could be based on their ability to effectively balance specificity and representativeness given the structure of visual information evoked by agent/environment interaction.

Chapter 9

General Discussion

This thesis investigates object recognition models from an active, embodied and situated perspective. Over several decades, many models have been proposed with the purpose of matching or understanding the performance of the human visual system. While significant progress has been made in the study of object recognition models for both purposes, the performance of artificial models is still far away from their counterpart in nature. Analogously, neural processes in object recognition are far from being completely understood.

To date, most models of object recognition are not studied under conditions where the active exploitation of movement and attentional mechanisms is made explicit. Certain models of object recognition have been proposed to exploit active approaches as a way of aiding the recognition process (Arbel and Ferrie, 2002, 2001). Other models have exploited embodied approaches to object recognition using computer vision based models in robot based applications(Andreasson and Duckett, 2003; Gvozdjak and Li, 1998). However, in these applications the focus is on achieving a specific task, rather than gaining an understanding of visual processes in natural systems. Biologically inspired models have been proposed to resemble the visual systems of insects or primates, however, usually these models are studied in a static or isolated way, rather than by taking into account active approaches. As all visual systems in nature are active and situated, the goal of this thesis was to study object recognition as an situated and active process.

In order to study object recognition models from an active, embodied and situated perspective, the first step was a comparison of two models of object recognition using an attentional mechanism, through which it was demonstrated that a simple V1-like model could perform as well as (or even better than) different implementations of a complex hierarchical model, the HMAX, in realistically noisy conditions. In order to provide the mechanisms that allow the simple model to outperform the HMAX in a more embodied scenario, an exploration of controllers for simulated agents was carried out. These simulated agents were designed initially using simple visual systems which were gradually increased in complexity in order to isolate the important processes involved. The active control of such visual systems was only feasible when the visual information being simulated was not significantly complex, so that the evolutionary techniques used could be applied in reasonable time scales. Based on these restrictions, an exploration of the exploitation of movement to improve object recognition in an active and embodied perspective was proposed. This exploration had two main goals, first, it was demonstrated that a simple model could actively exploit variations in the visual information imposed by simple movement strategies and, second, the conditions in which this model better exploits this variation were characterised. The exploitation of variations imposed by movement was not only considered in space but also in time. It was shown that the temporal structure imposed by movement could be exploited by the RBF model to help the object recognition processes in embodied mobile agents. To conclude this thesis, the validity of the results obtained in simulated experiments was validated in the real world using a robot to acquire visual information using a panoramic camera.

9.1 Summary

9.1.1 Chapter 3

In chapter 3, a thorough study of two models, the HMAX and RBF, was presented. The primary goal of this chapter was to see how the models fared on a static task in realistic conditions, and specifically to see if an active mechanism could reduce the necessary complexity of an object recognition model. In doing so, the chapter also satisfied its secondary goal which was to gain an understanding of the models in various conditions to underpin the subsequent research.

In order to do this, the performance of the HMAX and RBF models was compared in two experiments under different conditions. In the first experiment, the performance of these two models was contrasted with some state-of-the-art computer vision models using the COIL-100 object images database for a general purpose object recognition task. In this evaluation it was shown that the performance of the HMAX model was as good as the state-of-the-art models. In contrast, the performance of the RBF model was poor in comparison with the rest of the models.

In the second experiment, the performance of the HMAX and RBF model was compared under a series of variations in translation and scale using an attentional mechanism. In this experiment it was demonstrated that when an attentional mechanism is considered, a simple V1-like model, such as the RBF, can sometimes outperform the HMAX model in object recognition tasks that require translation and scale invariances. In particular, it was shown that as long as the focus of attention was maintained close enough to the object in the visual field, the RBF model outperforms the HMAX model. Through the experiments in this chapter, a deeper understanding of the hierarchical architecture of a complex model of the ventral pathway in the visual cortex was also gained.

9.1.2 Chapter 4

Chapter 4 was divided in two experiments. In the first experiment, it was shown that by restricting the panoramic sensors to directional sensors in an embodied agent using a simple visual system, the complexity of the controller required to perform simple object discrimination tasks can be reduced. This restriction requires an active perception approach in which the exploitation of movement becomes necessary in order to perform behavioural object discrimination. This part of the chapter it was focused on the exploration of controllers in order to see if they could provide active mechanisms such as those used in the previous chapter, so that a simple model would be able to outperform the HMAX in embodied visually guided agents. However, finding such controllers for an agent using complex visual information was a difficult task. In order to approach this problem, the methodology followed in this chapter proposed to find controllers for simple visually guided agents first, and gradually increase the complexity of the visual system. A hierarchy can be envisaged in which controllers are initially evolved in simple simulations and then are incrementally refined in progressively more complex simulations until final deployment in a real world environment. In addition, rich simulations offer the possibility of exploring detailed agent-environment interactions which do not exist in real-world situations, thereby supplying potentially valuable comparison conditions for understanding embodied visual systems. This methodology provides useful insights about tools for complex visual simulations to study or build visually guided agents using an ER approach.

9.1.3 Chapter 5

In chapter 5, the exploitation of simple movements is proposed to improve object recognition with a simple model. This proposition was based on the idea that, if simple movement strategies can be exploited by a simple model using attentional mechanisms so that it can perform object recognition reliably, then the complexity of the required controllers can be reduced. Therefore, in this chapter an active comparison of the models is carried out, where different trajectories were used to train and test the models.

This comparison showed that the models exploited the variation in the visual information imposed by simple movement strategies. It was demonstrated that by having multiple training views, the RBF increases its robustness to noise in the visual system. Furthermore, the variation in scale and rotation in the training views increases the discriminability of the RBF model due to its high specificity. In contrast, the discriminability of the HMAX is reduced due to its high generalisation. The difference in model activity when views were collected using different trajectories during training and testing suggests that there could be optimal movements for increasing the object recognition performance of these models. This concept is important because it proposes another way of studying the problem of finding suitable models for object recognition in autonomous mobile agents. This approach consists of exploiting movement from an active and situated perspective using a simple model of object recognition. The importance of the visual information acquisition process is stressed and it is shown that, with simple movement strategies, the complexity of the controller required can be reduced.

9.1.4 Chapter 6

In chapter 6, the conditions under which the models exploit variation in visual information following simple movement strategies is studied. In this chapter it was shown that, in conditions where significant variation in rotation and scale is provided during training, the RBF model outperforms the HMAX model as long as both models are provided with an attentional mechanism. It was demonstrated that such variation of rotation and scale can be provided by simple movement strategies. Additionally, it was also shown that the RBF model can exploit temporal structure in visual information imposed by movement. In these experiments it was further shown that when objects are difficult to discriminate from multiple points of view (for example in the discrimination of two objects that are very similar from most of points of view), the temporal information can be exploited by the RBF model to improve the recognition performance, when the variation in the visual information is similar during training and testing.

9.1.5 Chapter 7

In chapter 7, experiments were carried out using a gantry robot to collect visual information in the real world. The goal of this chapter was to test the models to see if the results from simulations could transfer to the real world. This goal was achieved, as the experiments carried out in this chapter validate the results and predictions of the previous simulated experiments. In particular, it was demonstrated that variation in the visual information can be exploited by the RBF model using simple movement strategies, when an attentional mechanism is considered. This variation in visual information is determined by the movement strategies and, it shapes particular regions in the arena where the performance of the model is better than when evaluated in the rest of the arena. Therefore, can be optimal regions for recognition performance of the model when enough variation in scale and rotation is provided during training. Furthermore, it was demonstrated that the temporal structure imposed by movement can also be exploited by the RBF model in real world conditions. In this case, it was also shown that the variation in the temporal structure shapes regions where the performance of the model is better. The results in this chapter are important because they demonstrate that the RBF model can be used to perform object recognition in an active autonomous mobile robot in real world conditions.

9.1.6 Chapter 8

In chapter 8, we analyse whether 'activeness' can compensate for reduction in complexity of the RBF model in a real robot. Our hypothesis was that by reducing the complexity of the RBF model, the specificity of the processed views by the reduced versions of the RBF will be reduced, and as a consequence, their performance. We reduced the complexity of the RBF model by decreasing the number of filter sizes and orientations that the model used to process the visual information. To regain performance, we proposed two methods that exploit regularities in incoming visual information to select training views through movement. The first method uses a criterion based on the similarity to assure that an 'acceptable level of specificity' is kept in the training views. The second method uses a 'representativeness' criterion so that the selected training views are representatives of the distribution of images potentially acquired during the training and testing phases. Additionally, we established a baseline performance for the RBF models by selecting the training views at fixed intervals without any active selection criteria. We found that, even though using these methods to select the training views increases the performance of the reduced RBF models in comparison with the baseline performance, each criterion on its own is not sufficient to fully recover the performance of the complete version of the RBF model. However, we conclude that by exploiting movement during training and tracking criteria, the performance of RBF models can be improved, suggesting that active strategies for object recognition could be based on their ability to effectively balance specificity and representativeness given the structure of visual information evoked by agent/environment interaction.

9.2 Future work

There are several interesting avenues that could be followed for the extension and exploration of ideas raised throughout this thesis. One direction would be to make the processes of the system more active and adaptive. For example, we know that the accuracy of a simple model does not need to be perfect from any point in the arena to successfully recognise objects in an active mobile agent, rather, it only needs to be accurate in the proximity to the objects. Therefore, the exploration of simple controllers using features, such as optic flow, to perform simple movement strategies to collect views during training and testing would be interesting. An initial idea could be to perform object approaching until the agent is within regions where the performance of the model is good (see advantageous regions in chapter 7) and then use an adaptive movement strategy modulated by the similarity between the views or the recognition signals from the model. During training, a similarity measure between the views could be used such that a new training view is collected by the agent every time a threshold is reached in the similarity between the current and previous views.

Although the final goal of this thesis is not the optimisation of existing models, several extensions of the work presented in this thesis can be envisaged for this purpose. For example, given that the overall performance of the system is influenced by the accuracy of the selection process of blobs (see chapter 7), the optimisation of these processes is important. One way of optimising the selection criteria in the BDM could be using an adaptive mechanism to change or combine different selection criteria.

Another optimisation could be carried out by exploring different advantageous regions in the arena to improve the recognition performance. This is particularly the case for the exploitation of the temporal information using the RBF model (DBCV) in real world conditions. In the experiments previously presented, this region was arbitrarily selected based on the performance of the model using single view presentations (SVP). A different region for the DBCV case could represent an improvement in the performance of the model when restricted to that region (see chapter 7).

Finally for chapter 8, given that RBF_C has lost spatial filters, one might predict that recovery of performance could be achieved by increasing the range of distances over which training views were acquired. Despite some recovery, we did not find that this was generally the case. One possible reason for this outcome is that active/adaptive strategies are unlikely to compensate for loss of rotational filters (thus explaining differences between

RBF_B and RBF_C). Future work would advisably involve explicit image rotation during training views selection. Additional and a priori unknown criteria would be needed to guide selection of rotated images; it is possible that a variation of the neighborhood method could be useful here.

9.3 Final conclusions

Naturally, it is common to think about object recognition in mobile agents as a process that requires significant invariances in scale, translation, illumination, and other variable conditions. However, in this work it has been demonstrated that by posing this problem from an active vision perspective in an embodied and situated agent, a simple biologically inspired model can be employed successfully for object recognition tasks in the real world. This kind of approach raises several questions in the modelling of object recognition, particularly for biologically inspired models, given that a significant amount of the complexity in most of these models is invested towards the achievement of position-independent object recognition. However, the study of the exploitation of variations in the visual information imposed by movement from an active, embodied, and situated approach, such as in this work, invites us to consider the possibility of using attentional mechanisms and the exploitation of movement in the modelling of object recognition processed in the brain. The idea that the recognition process is not a position-independent processes has been recently raised in the neuroscience community. In (Kravitz et al., 2008), they conclude that there is little evidence to support position-independent object recognition. In robotics, Spier (2004); Suzuki (2007) use the active vision approach to reduce the complexity of the visual sensors to perform behavioural tasks. They stress that behaviour can make up for bad sensors or bad vision. Analogously, object recognition is also an active process that is not restricted to be accurate from static single view perspectives but is a dynamic process that exploits movement in space and time.

Appendix

The values of total number of the correct guesses for the RBF versions using different training view selection methods and movement strategies in chapter 8 are shown in table A1.

Var	objects	Var	objects
A-T1-σ3.0	591 193 823 1951 862 364 1075	A-T2-σ3.0	1708 793 1350 718 575 291 991
A-T3-σ3.0	90 243 1240 1825 1150 333 123	A-T4-σ3.0	819 622 1916 336 1181 102 1169
A-T1-σ3.0-C	591 193 823 1951 862 364 1075	A-T2-σ3.0-C	1636 1252 1161 1197 728 675 1167
A-T3-σ3.0-C	81 234 1143 1843 465 369 127	A-T4-σ3.0-C	819 777 1163 1704 1325 486 1335
A-TS-σ1.8-I1A	50 118 30 35 27 1663 68	A-TS-σ1.8-I2A	111 462 136 313 85 2020 206
A-TS-σ1.8-I3A	240 511 357 571 204 2017 267	A-TS-σ1.8-I6A	527 1039 1199 756 937 1967 991
A-TS-σ1.8-I3C	840 926 1184 766 1490 1258 1296	A-TS-σ1.8-I6C	500 1206 1176 1186 1352 1144 1171
A-TS-σ1.8-TA	485 847 1191 859 885 782 972	A-TS-σ1.8-TB	202 742 1222 834 917 1794 850
A-TS-σ1.8-TC	492 553 457 1721 121 590 400		
A-TS-σ1.8-NA	675 1111 1220 717 1409 1271 1245	A-TS-σ1.8-NB	692 1167 1206 770 1244 1291 1246
A-TS-σ1.8-NC	674 1188 1205 826 1227 1375 1254		
B-T1-σ3.0	1447 127 348 1532 764 537 1300	B-T2-σ3.0	1114 473 1278 389 738 582 589
B-T3-σ3.0	575 27 1214 296 1547 6 164	B-T4-σ3.0	1002 14 1557 129 645 0 294
B-T1-σ3.0-C	1447 127 348 1532 764 537 1300	B-T2-σ3.0-C	1170 495 1163 768 718 731 531
B-T3-σ3.0-C	536 22 280 1320 1301 8 143	B-T4-σ3.0-C	1087 14 1263 737 700 0 304
B-TS-σ1.8-I1A	0 1391 1266 0 782 29 80	B-TS-σ1.8-I2A	287 194 1327 367 989 1446 303
B-TS-σ1.8-I3A	471 138 930 1701 898 49 743	B-TS-σ1.8-I6A	1125 247 1131 661 853 1446 827
B-TS-σ1.8-I3C	1507 67 1221 522 948 1226 688	B-TS-σ1.8-I6C	1414 255 1130 888 1010 696 602
B-TS-σ1.8-TA	1263 326 1033 888 1034 638 474	B-TS-σ1.8-TB	911 214 1013 756 1208 1272 611
B-TS-σ1.8-TC	963 124 1385 494 593 1249 424		
B-TS-σ1.8-NA	1367 306 1148 538 910 1251 677	B-TS-σ1.8-NB	1435 346 1145 663 703 1299 561
B-TS-σ1.8-NC	1427 336 1280 555 771 1175 607		
C-T1-σ3.0	1549 128 194 1401 710 368 1362	C-T2-σ3.0	1131 630 945 294 635 742 593
C-T3-σ3.0	587 95 991 142 367 1 1119	C-T4-σ3.0	640 39 1221 17 252 0 359
C-T1-σ3.0-C	1549 128 194 1401 710 368 1362	C-T2-σ3.0-C	1138 579 954 589 606 883 642
C-T3-σ3.0-C	570 87 220 969 1186 9 787	C-T4-σ3.0-C	648 39 1105 199 266 0 376
C-TS-σ1.8-I1A	0 15 27 0 1746 0 356	C-TS-σ1.8-I2A	367 113 650 31 604 1057 300
C-TS-σ1.8-I3A	323 107 260 1390 84 11 651	C-TS-σ1.8-I6A	802 180 978 516 442 1221 430
C-TS-σ1.8-I3C	1420 23 1186 492 772 1139 504	C-TS-σ1.8-I6C	701 243 846 825 432 528 291
C-TS-σ1.8-TA	1076 194 594 912 890 707 279	C-TS-σ1.8-TB	922 161 767 945 763 804 428
C-TS-σ1.8-TC	433 110 1861 20 443 6 274		
C-TS-σ1.8-NA	1209 195 1010 510 824 1119 530	C-TS-σ1.8-NB	1302 282 1050 674 641 1238 483
C-TS-σ1.8-NC	1253 301 1230 552 718 1060 490		

Table A1: Number of correct guesses for each object when using the RBF_A, RBF_B and RBF_C models. The names of the variables denote the version of the RBF used, the movement strategy used to collect the training views, the value of sigma employed in the classifier, whether the blobs were corrected or not (in case the T1, T2, T3 or T4 were emploted) and the method to select training views in case the spiral movement strategy was used. For example, C-TS-σ1.8-NA denotes RBF_C (C), using the spiral movement strategy (TS) with $\sigma = 1.8$, employing the neighbourhood method to select the training views with the RBF_A (NA). B-TS-σ1.8-I3C denotes RBF_B, using the spiral movement strategy (TS) with $\sigma = 1.8$, the interval based view selection method with interval of 3 in clockwise direction (I3C).

Bibliography

Aloimonos, Y., editor (1993). *Active Perception*. Erlbaum, Hillsdale, NJ.

Andreasson, H. and Duckett, T. (2003). Object recognition by a mobile robot using omnidirectional vision. In *Proc. Eighth Scandinavian Conference on Artificial Intelligence (SCAI 2003)*.

Arbel, T. and Ferrie, F. P. (2001). Entropy-based gaze planning. *Image and Vision Computing*, 19(11):779–786.

Arbel, T. and Ferrie, F. P. (2002). Interactive visual dialog. *Image and Vision Computing*, pages 639–646.

Arman, F. and Aggarwal, J. K. (1993). Model-based object recognition in dense-range images—a review. *ACM Comput. Surv.*, 25(1):5–43.

Bajcsy, R. (1988). Active perception. In *Proceedings of the IEEE, Special issue on Computer Vision*, volume 76(8).

Ballard, D. H. (1991). Animate vision. *Artificial Intelligence*.

Beer, R. (1995). On the dynamics of small continuous-time recurrent neural networks. *Adaptive Behavior*, 3(4):469–509.

Beer, R. D. (2000). Dynamical approaches to cognitive science. *Trends in Cognitive Sciences*, 4(3):91–99.

Beer, R. D. (2003). The dynamics of active categorical perception in an evolved model agent. *Adaptive Behavior*, 11(4):209–243.

Bekey, G. A. (2005). *Autonomous Robots: From Biological Inspiration to Implementation and Control*. The MIT Press.

Bermudez, E. (2007a). An account for a biologically inspired machine vision system. In *Student Meeting of the British Machine Vision Association (BMVA)*.

Bermudez, E. (2007b). A biologically inspired solution for an evolved simulated agent. In *Proc. of the Genetic and Evolutionary Computation Conference (GECCO)*. ACM.

Bermudez, E., Philippides, A., and Seth, A. (2008). Movement strategies for learning in visual recognition. In *Proc. XI International Conference on Artificial Life (ALife XI)*.

Bermudez, E. and Seth, A. (2007). Simulations of simulations in evolutionary robotics. In Almeida e Costa, F., Mateus Rocha, L., Costa, E., Harvey, I., and Coutinho, A., editors, *Proc. European Conference of Artificial Life (ECAL)*, pages 796–806. Springer-Verlag.

Bermudez-Contreras, E., Buxton, H., and Spier, E. (2008). Attention can improve a simple model for visual object recognition. *Image and Vision Computing*, 26:776–787.

Bernardino, A. and Santos-Victor, J. (2002). A binocular stereo algorithm for log-polar foveated systems. In *2nd Workshop on Biological Motivated Computer Vision, BMCV 2002*, pages 127–136.

Bianco, G., Zelinsky, A., and Lehrer, M. (2000). Visual landmark learning. In *Proceedings of the International Conference on Intelligent Robots and Systems, (IROS 2000)*, pages 227 – 232.

Biederman, I. (1987). Recognition-by-components: a theory of human image understanding. *Psychol Rev*, 94(2):115–147.

Bishop, C. M. (1996). *Neural Networks for Pattern Recognition.* Oxford University Press, USA, 1 edition.

Booth, M. and Rolls, E. (1998). View-invariant representations of familiar objects by neurons in the inferior temporal visual cortex. *Cerebral Cortex*, 8:510–523.

Borotschnig, H., Paletta, L., Prantl, M., and Pinz, A. (2000). Appearance-based active object recognition. *Image and Vision Computing*, 18:715–727.

Brooks, R. A. (1992). Artificial life and real robots. In *Proceedings of the First European Conference on Artificial Life*, pages 3–10. MIT Press.

Cartwright, B. and Collett, T. (1983). Landmark learning in bees: Experiments and models. *Journal of Comparative Physiology*, 151:521–543.

Cedras, C. and Shah, M. (1995). Motion-based recognition: a survey. *Image and Vision Computing*, 13(2):129–155.

Chen, J.-H. and Chen, C.-S. (2004). Object recognition based on image sequences by using inter-feature-line consistencies. *Pattern Recognition*, 37(9):1913–1923.

Cliff, D. and Miller, G. F. (1996). Co-evolution of pursuit and evasion II: Simulation methods and results. In Maes, P., Mataric, M. J., Meyer, J.-A., Pollack, J. B., and Wilson, S. W., editors, *From animals to animats 4*, pages 506–515, Cambridge, MA. MIT Press.

Collett, T. S. and Rees, J. A. (1997). View-based navigation in hymenoptera: multiple strategies of landmark guidance in the approach to a feeder. *Journal of Comparative Physiology A: Neuroethology, Sensory, Neural, and Behavioral Physiology*, 181(1):47–58.

Deco, G. and Lee, T. S. (2004). The role of early visual cortex in visual integration: a neural model of recurrent interaction. *European Journal of Neuroscience*, 20:1089–1100.

Duchon, A., Warren, W., and L., P. K. (1998). Ecological robotics. *Special Issue on Biologically Inspired Models of Spatial Navigation*, 6(3).

Duvdevani-Bar, S., Edelman, S., Howell, A. J., and Buxton, H. (1998). A similarity-based method for the generalization of face recognition over pose and expression. *Proc. 3rd IEEE International Conference on Automatic Face & Gesture Recognition (FG'98)*, pages 118–123. Tara, Japan.

Edelman, S. (1997). Computational theories of object recognition. *Trends in Cognitive Sciences*, 1:296–304.

Edelman, S. and Duvdevani-Bar, S. (1997). A model of visual recognition and categorization. *Phil. Trans. R. Soc. Lond. B*, 352(1358):1191–2002.

Fergus, R., Perona, P., and Zisserman, A. (2003). Object class recognition by unsupervised scale-invariant learning. In *Proceedings of CVPR*, volume 2, pages 264–271.

Fernald, R. (2004). Evolving eyes. *Int. J. Dev. Biol.*, 48:701–705.

Findlay, J. M. and Gilchrist, I. D. (2003). *Active vision: the psychology of looking and seeing.* Oxford University Press, New York.

Floreano, D., Godjevac, J., Martinoli, A., Mondada, F., and Nicoud, J. (1998). Design, Control, and Applications of Autonomous Mobile Robots. In Tzafestas, S. G., editor, *Advances in Intelligent Autonomous Agents.* Kluwer Academic Publishers, Boston. Part 2, Chapter 8, p. 159-186.

Floreano, D., Kato, T., Marocco, D., and Sauser, E. (2004). Coevolution of active vision and feature selection. *Biological Cybernetics*, 90:218–228.

Funahashi, K.-I. and Nakamura, Y. (1993). Approximation of dynamical systems by continuous time recurrent neural networks. *Neural Netw.*, 6(6):801–806.

Gvozdjak, P. and Li, Z. N. (1998). From nomad to explorer: active object recognition on mobile robots. *Pattern recognition*, (6):773–790.

Harvey, I., Husbands, P., and Cliff, D. (1994). Seeing the light: aritificial evolution, real vision. In Cliff, D., Husbands, P., Meyer, J., and Wilson, S., editors, *From animals to animats III*, pages 392–401.

Harvey, I., Paolo, E. A. D., Tuci, E., Wood, R., and Quinn, M. (2005). Evolutionary robotics: A new scientific tool for studying cognition. *Artificial Life*, 11:79–98.

Haykin, S. (1994). *Neural Networks: A Comprehensive Foundation.* Macmillan, New York.

Heisele, B., Serre, T., Pontil, M., Vetter, T., and Poggio, T. (2002). Categorization by learning and combining object parts. In *Proceedigs of Neural Information Processing Systems (NIPS)*, Vancouver, Canada.

Howell, J. and Buxton, H. (1995). Receptive fields functions for face recognition. In *Proc. 2nd International Workshop on Parallel Modelling of Neural Operators for Pattern Recognition*, pages 221–226.

Hung, C., Kreiman, G., Poggio, T., and DiCarLo, J. (2005). Fast readout of object identity from macaque inferior temporal cortex. *Science*, 310:863–866.

Itti, L. and Koch, C. (2001). Computational modelling of visual attention. *Nature*, 2:194–203.

Jakobi, N. (1998). The minimal simulation approach to evolutionary robotics. In Gomi, T., editor, *Proceedings of Evolutionary Robotics - From Intelligent Robots to Artificial Life.* AAI Books.

Kravitz, D. J., Vinson, L. D., and Baker, C. I. (2008). How position dependent is visual object recognition? *Trends in Cognitive Sciences*, 12(3):114–122.

Land, M. and Fernald, R. (1992). The evolution of eyes. *Annu. Rev. Neuroscience*, 15:1–29.

Land, M. F. and Nilsson, D. E. (2002). *Animal Eyes.* Oxford University Press.

Laughlin, S. and Sejnowski, T. (2003). Communication in neuronal networks. *Science*, 301(5641):1870–1874.

Lazebnik, S., Schmid, C., and Ponce, J. (2006). Beyond bags of features: Spatial pyramid matching for recognizing natural scene categories. In *In proceedings of IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, volume 2, pages 2169–2178.

Lee, T. S. (2003). Analysis and synthesis of visual images in the brain: evidence for pattern theory. In Olver, P. and Tannenbaum, A., editors, *Image Analysis and Higher Level Vision*, Lecture notes in Mathematics and its Application, pages 87–106. Springer-Verlag.

Lehrer, M. and Bianco, G. (2000). The turn-back-and-look behaviour: bee versus robot. *Biological cybernetics*, 83(3):211–229.

Leung, B. (2004). Component-based car detection in street scene images. Master's thesis, EECS, MIT.

Liese, A., Polani, D., and Uthmann, T. (2001). A study of the simulated evolution of the spectral sensitivity of visual agent receptors. *Special Issue on Evolution of Sensors in Nature, Hardware and Simulation. Artificial Life*, 7(2):99–124.

Logothetis, N., Pauls, J., Bulthoff, H., and Poggio, T. (1994). View dependent object recognition by monkeys. *Curr. Biol.*, 4:401–414.

Logothetis, N., Pauls, J., and Poggio, T. (1995). Shape representation in the inferotemporal cortex of monkeys. *Curr. Biol.*, 5:552–563.

Lowe, D. G. (2004). Distinctive image features from scale-invariant keypoints. *International Journal of Computer Vision*, 60(2):91–110.

Marr, D. and Nishihara, H. K. (1978). Representation and recognition of the spatial organization of three-dimensional shapes. *Proceedings of the Royal Society of London. Series B, Biological Sciences*, 200(1140):269–294.

Meger, D., Forssen, P., Lai, K., Helmer, S., Mccann, S., Southey, T., Baumann, M., Little, J., and Lowe, D. (2008). Curious george: An attentive semantic robot? *Robotics and Autonomous Systems*, 56(6):503–511.

Mikolajczyk, K., Leibe, B., and Schiele, B. (2005). Local features for object class recognition. In *ICCV '05: Proceedings of the Tenth IEEE International Conference on Computer Vision*, pages 1792–1799, Washington, DC, USA. IEEE Computer Society.

Mokhtarian, F. and Abbasi, S. (2005). Robust automatic selection of optimal views in multi-view free-form object recognition. *Pattern Recognition*, 38(7):1021–1031.

Murase, H. and Nayar, S. (1995). Visual learning and recognition of 3-d objects from appearance. *International Journal of Computer Vision*, 14:5–24.

Mutch, J. and Lowe, D. G. (2006). Multiclass object recognition with sparse, localized features. *Computer Vision and Pattern Recognition, 2006 IEEE Computer Society Conference on*, 1:11–18.

Nayar, S. K., Nene, S. A., and Murase, H. (1996). Real-time 100 object recognition system. In *Robotics and Automation, 1996. Proceedings., 1996 IEEE International Conference on*, volume 3, pages 2321–2325 vol.3.

Nene, S., Nayar, S., and Murase, H. (1996). Columbia object image library: Coil.

Nolfi, S. and Floreano, D. (2002). Synthesis of autonomous robots through evolution. *Trends in Cognitive Science*, 6(1):31–36.

Nolfi, S. and Marocco, D. (2000). Evolving visually-guided robots able to discriminate between different landmarks. In *In: J-A Meyer, A. Berthoz, D. Floreano, H.L. Roitblat, and S.W. Wilson (eds.) From Animals to Animats 6. Proceedings of the VI International Conference on Simulation of Adaptive Behavior*. MIT Press.

Orr, M. (1996). Introduction to radial basis functions networks. Technical report, Centre of Cognitive Science, University of Edinburgh.

Orr, M. (1998). Optimising the widths of radial basis functions. In *Fifth Brazilian Symposium on Neural Networks*.

Orr, M., Hallam, J., Murray, A., and Leonard, T. (2000). Assessing rbf networks using delve. *International Journal of Neural Systems*, 10:397–415.

Paletta, L., Rome, E., and Buxton, H. (2005). Attention architectures for machine vision and mobile robots. *Neurobiology of Attention*, pages 642–648.

Palmer, S. (1999). *Vision Science – photons to phenomenology.* The MIT Press.

Palmeri, T. J. and Gauthier, I. (2004). Visual object understanding. *Nature reviews. Neuroscience*, 5(4):291–303.

Peters, G. (2000). Theories of three-dimensional object perception: A survey. *Recent research developments in pattern recognition*, 1:179–197.

Pfeifer, R. and Scheier, C. (1999). *Understanding Intelligence*. MIT Press.

Pinto, N., Cox, D. D., and Dicarlo, J. J. (2008). Why is real-world visual object recognition hard? *PLoS Computational Biology*, 4(1):151–156.

Poggio, T. and Edelman, S. (1990). A network that learns to recognize 3-d objects. *Nature*, 343:263–266.

Rawlings, J. O., Pantula, S. G., and Dickey, D. A. (2001). *Applied Regression Analysis: A Research Tool (Springer Texts in Statistics)*. Springer.

Riesenhuber, M. and Poggio, T. (1999a). Are cortical models really bound by the "binding problem"? *Neuron*, 24:87–93.

Riesenhuber, M. and Poggio, T. (1999b). Hierarchical models of object recognition in cortex. *Nature Neuroscience*, 2(11):1019–1025.

Riesenhuber, M. and Poggio, T. (2000). Models of object recognition. *Nature Neuroscience*, 3:1199–1204.

Riesenhuber, M. and Poggio, T. (2003). *The Visual Neurosciences*, chapter How Visual Cortex Recognizes Objects: The Tale of the Standard Model. MIT Press.

Rosh, E. (1973). Natural categories. *Cognitive Psychology*, 4:328–350.

Roth, D., Yang, M., and Ahuja, N. (2002). Learning to recognize 3d objects. *Neural Computation*, 14(5):1071–1104.

Roy, S., Chaudhury, S., and Banerjee, S. (2004). Active recognition through next view planning: a survey. *Pattern Recognition*, 37(3):429–446.

Saffiotti, A. (1998). Handling uncertainty in control of autonomous robots. In *Applications of Uncertainty Formalisms*, pages 198–224. Springer Verlag.

Schneider, R. and Riesenhuber, M. (2002). A detailed look ar scale and translation invariance in a hierarchical neural model of visual object recognition. Technical Report Memo 2002-011, Massachusetts Institute of Technology. ftp://publications.ai.mit.edu/ai-publications/2002/AIM-2002-011.pdf.

Schneider, R. and Riesenhuber, M. (2004). On the difficulty of feature-based attentional modulations in visual object recognition: A modeling study. Technical Report Memo 2004-004, Massachusetts Institute of Technology. ftp://publications.ai.mit.edu/ai-publications/2004/AIM-2004-004.pdf.

Serre, T., Kouh, M., Cadieu, C., Knoblich, U., Kreiman, G., and Poggio, T. (2004). A new biologically motivated framework for robust object recognition. Cbcl paper #243/ai memo #2004-026, Massachusetts Institute of Technology, MIT, Cambridge, MA.

Serre, T., Kouh, M., Cadieu, C., Knoblich, U., Kreiman, G., and Poggio, T. (2005a). A theory for object recognition: Computations and circuits in the feedforward path of the ventral visual stream in primate visual cortex. Cbcl paper #259/ai memo #2005-036, Massachusetts Institute of Technology.

Serre, T., Wolf, L., and Poggio, T. (June, 2005b). Object recognition with features inspired by visual cortex. In *Proceedings of 2005 IEEE Computer Society Conference on Computer Vision and Patter Recognition (CVPR).*

Spier, E. (2004). Behavioural categorisation: Behaviour makes up bad vision. *Artificial Life IX: Proceedings of the Ninth International Conference on the Simulation and Synthesis of Living Systems*, pages 133–139.

Stringer, S. M. and Rolls, E. T. (2002). Invariant object recognition in the visual system with novel views of 3d objects. *Neural Comp.*, 14(11):2585–2596.

Suzuki, M. (2007). *Enactive robot vision.* PhD thesis, Lausanne.

Teynor, A., Rahtu, E., Setia, L., and Burkhardt, H. (2006). Properties of patch based approaches for the recognition of visual object classes. In *DAGM06*, pages 284–293.

Ullman, S. (1996). *High-level Vision. Object recognition and visual cognition.* The MIT Press.

Wallis, G. and Bulthoff, H. (1999). Learning to recognize objects. *Trends in Cognitive Sciences*, 3(1):22–31.

Walther, D., Itti, L., Riesenhuber, M., Poggio, T., and Koch, C. (2002). Attentional selection for object recognition – a gentle way. In Lee, S.-W., Buelthoff, H., and Poggio, T., editors, *Second IEEE International Workshop*, Biologically Motivated Computer Vision, pages 387–397.

Wang, G., Zhang, Y., and Fei-Fei, L. (2006). Using dependent regions for object categorization in a generative framework. In *Proc. IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pages 1597–1604.

Watanabe, M., Takeda, N., and Onoguchi, K. (1996). A moving object recognition method by optical flow analysis. In *Proc. International Conference of Pattern Recognition (ICPR'96)*, volume 1, page 528. IEEE.

Webb, B. (1996). A robot cricket. *Scientific American*, 275(6):94–99.

Weber, M., Welling, M., and Perona, P. (2000). Unsupervised learning of models for object recognition. In *Proceedings of ECCV*.

Yantis, S., editor (2000). *Visual perception*. Psychology Press.

Young, D. (2000). First-order optic flow and the control of action. In *Proceedings of the European Conference on Visual Perception (ECVP2000)*.

Zhang, H., Berg, A. C., Maire, M., and Malik, J. (2006a). Svm-knn: Discriminative nearest neighbor classification for visual category recognition. In *Proc. IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, volume 2, pages 2126–2136.

Zhang, H., Berg, A. C., Maire, M., and Malik, J. (2006b). Svm-knn: Discriminative nearest neighbor classification for visual category recognition. In *IEEE Computer Society Conference on Computer Vision and Pattern Recognition - Volume 2 (CVPR 06)*, pages 2126–2136.

Zhaoping, L. (2002). A saliency map in primary visual cortex. *Trends in Cognitive Science*, 6(1):9–16.